

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



SISTEMA MULTIAGENTE PARA EL MANEJO ÓPTIMO DE UN CULTIVO
EMPLEANDO LA METODOLOGÍA PROMETHEUS

POR

DEIBY ANDRÉS PARRA GAVIRIA

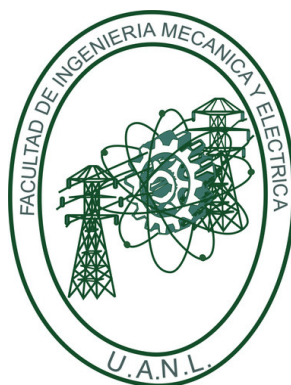
EN OPCIÓN AL GRADO DE MAESTRÍA EN CIENCIAS
EN INGENIERÍA DE SISTEMAS

MAYO, 2017

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



SISTEMA MULTIAGENTE PARA EL MANEJO
ÓPTIMO DE UN CULTIVO EMPLEANDO LA
METODOLOGÍA PROMETHEUS

POR

DEIBY ANDRÉS PARRA GAVIRIA

EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

MAYO 2017

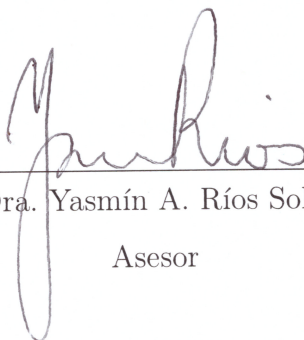
Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Sistema multiagente para el manejo óptimo de un cultivo empleando la metodología Prometheus», realizada por el alumno Deiby Andrés Parra Gaviria, con número de matrícula 1770701, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis



Dra. Yasmín A. Ríos Solís

Asesor



Dr. Romeo Sánchez Nigeda

Revisor



Dr. Néstor Miguel Cid García

Revisor

Vo. Bo.



Dr. Simón Martínez Martínez

Subdirector de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, mayo 2017

Las Tres leyes de la robótica:

- 1. Un robot no puede dañar a un ser humano ni, por inacción, permitir que un ser humano sufra daño.*
- 2. Un robot debe obedecer las órdenes dadas por los seres humanos excepto cuando tales órdenes entren en conflicto con la Primera Ley.*
- 3. Un robot debe proteger su propia existencia hasta donde esta protección no entre en conflicto con la Primera o Segunda Ley.*

Isaac Asimov

ÍNDICE GENERAL

Agradecimientos	XIV
1. Resumen	1
2. Introducción	3
2.1. Descripción del problema	3
2.2. Introducción	4
2.3. Objetivo	6
2.3.1. Objetivos específicos	6
3. Marco Teórico	7
3.1. Sistemas de agentes	7
3.2. Metodología Prometheus	10
3.3. Agricultura de precisión	11
3.3.1. ¿Qué es un vehículo aéreo no tripulado (VANT)?	12
3.3.2. Uso de drones en la agricultura	12
3.4. Heurísticas y metaheurísticas	13

3.4.1. Búsqueda local	13
3.4.2. Ascenso de colinas	14
3.4.3. Búsqueda Tabú	16
4. Modelación del problema	18
4.1. Entrada de datos	19
4.2. Salida de datos	19
4.3. Función objetivo	20
4.4. Restricciones	20
5. Diseño del Sistema Multiagente	21
5.1. Modelado del sistema	22
5.1.1. Presentación del problema	22
5.1.2. Fase especificación del sistema	23
5.1.3. Fase diseño arquitectónico	28
5.1.4. Fase diseño detallado	31
6. Algoritmos implementados	33
6.1. Búsqueda local	33
6.2. Búsqueda ascenso de colinas (Hill Climbing)	34
6.3. Búsqueda Tabú	36
7. Metodología	38

7.1. Equipo computacional	39
7.2. Implementación computacional	39
7.3. Supuestos considerados	39
7.4. Modelos planteados	40
7.4.1. Primer modelo	41
7.4.2. Segundo modelo	41
7.4.3. Tercer modelo	42
7.5. Diseño de experimentos	42
8. Experimentación	43
8.1. Primer modelo	43
8.2. Segundo modelo	46
8.3. Tercer modelo	48
8.4. Análisis general	50
9. Conclusiones	56
9.1. Conclusiones	56
9.2. Trabajo futuro	57
A. Tablas diseño sistema multiagente	59
A.1. Fase especificación del sistema	60
A.1.1. Objetivos	60

A.1.2. Roles	62
A.1.3. Escenarios	63
A.1.4. Percepciones	64
A.1.5. Acciones	68
A.2. Fase diseño arquitectónico	71
A.2.1. Identificación de los agentes	71
A.2.2. Datos	74
A.2.3. Protocolos de comunicación	74
A.2.4. Mensajes	76
A.3. Fase diseño detallado	78
A.3.1. Capacidades	78
A.3.2. Planes	80

ÍNDICE DE FIGURAS

3.1. Modelo de agente	9
5.1. Simbología del sistema	23
5.2. Análisis Preliminar	24
5.3. Escenarios	25
5.4. Diagrama Recargar Escenario	25
5.5. Diagrama optimizar escenario	27
5.6. Objetivos	27
5.7. Roles del sistema	28
5.8. Definición de agentes	29
5.9. Acoplamiento de datos	30
5.10. Interacciones entre agentes	30
5.11. Arquitectura Global	31
5.12. Capacidad agentes	32
8.1. Costo modelo 1 tamaño 20	44

8.2. Eficiencia modelo 1 tamaño 20	44
8.3. Costo modelo 1 tamaño 30	45
8.4. Eficiencia modelo 1 tamaño 30	45
8.5. Costo modelo 2 tamaño 20	46
8.6. Eficiencia modelo 2 tamaño 20	47
8.7. Costo modelo 2 tamaño 30	47
8.8. Eficiencia modelo 2 tamaño 30	48
8.9. Costo modelo 2 tamaño 20	48
8.10. Eficiencia modelo 3 tamaño 20	49
8.11. Costo modelo 3 tamaño 30	49
8.12. Eficiencia modelo 3 tamaño 30	50
8.13. Histograma costo vs modelo y tamaño	51
8.14. interacción costo vs modelo vs tamaño	52
8.15. Histograma residuos trabajo vs modelo y tamaño	53
8.16. interacción trabajo vs modelo vs tamaño	53
8.17. Costo vs tamaño y modelo	54
8.18. Trabajo vs tamaño y modelo	55

ÍNDICE DE TABLAS

8.1. ANOVA costo vs modelo y tamaño	51
8.2. ANOVA Trabajo vs modelo y tamaño	52
A.1. Recargar batería	60
A.2. Controlar la humedad	61
A.3. Controlar la humedad	61
A.4. Controlar las plagas	62
A.5. Rol Optimizar	62
A.6. Rol Bateria	63
A.7. Recharge scenario	63
A.8. Optimizar Escenario	64
A.9. Percepción nivel de batería	65
A.10.Percepción otro dron	65
A.11.Percepción base para recargar	66
A.12.Percepción movimientos disponibles	66
A.13.Percepción zona para tratamiento	67

A.14.Percepción posición actual	67
A.15.Percepción tamaño del terreno	68
A.16.Acción avanzar a zona para trabajar	68
A.17.Acción optimizar área	69
A.18.Acción moverse por el terreno	69
A.19.Acción mover a base	70
A.20.Acción recargar la batería	70
A.21.Agente drone1	71
A.22.Agente drone2	72
A.23.Agente drone3	73
A.24.Protocolo irrigar	74
A.25.Protocolo abonar	75
A.26.Protocolo fumigar	76
A.27.Mensaje irrigar	76
A.28.Respuesta irrigar	77
A.29.Mensaje abonar	77
A.30.Respuesta abonar	77
A.31.Mensaje fumigar	78
A.32.Respuesta fumigar	78
A.33.Capacidad recargar	79

A.34.Capacidad optimizar	80
A.35.Plan revisar(zona)	81
A.36.Plan irBase	82
A.37.Plan trabajar	83
A.38.Plan parar	84

AGRADECIMIENTOS

Gracias a mi madre por su esfuerzo y dedicación, a mi hermano por su compañía y amistad y a todas aquellas personas que hicieron posible de una u otra forma la consecución de este logro.

Deiby Andrés Parra Gaviria
San Nicolás de los Garza, Nuevo León. 2017.

CAPÍTULO 1

RESUMEN

Deiby Andrés Parra Gaviria

Candidato para el grado de Maestria en Ciencias
en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

SISTEMA MULTIAGENTE PARA EL MANEJO
ÓPTIMO DE UN CULTIVO EMPLEANDO LA
METODOLOGÍA PROMETHEUS

Número de páginas: 88.

OBJETIVOS Y MÉTODO DE ESTUDIO: Diseñar un sistema multiagente empleando la metodología Prometheus para el manejo eficiente de un cultivo.

CONTRIBUCIONES Y CONCLUSIONES: Se tiene una serie de vehículos aéreos no tripulados (drones) y un área agrícola en la cual es necesario optimizar sus condiciones

físico-químicas. Los drones se emplean para realizar las tareas (riego, fertilización y detección de plagas) mediante el recorrido del terreno y uso de sensores. Los drones presentan ciertas limitaciones: capacidad limitada de energía, altura máxima de vuelo, rango de alcance, entre otras; por lo cual se hace necesario encontrar las condiciones adecuadas para la realización de las labores.

Para solucionar este problema se desarrolló un sistema multiagente en combinación de algoritmos de búsqueda para mejorar el movimiento de los drones a través del área definida. Al emplear los sistemas multiagentes, se obtiene un mejor desempeño en el campo de trabajo, ya que los drones tienen una comunicación permanente entre ellos, la información se puede procesar en tiempo real y trabajan mancomunados por un mismo objetivo. Las decisiones se toman al instante, lo que representa un ahorro de costos tanto para los drones como para el productor.

Firma del asesor: _____

Dra. Yasmín A. Ríos Solís

CAPÍTULO 2

INTRODUCCIÓN

*Temo el día en que la tecnología sobrepase nuestra humanidad;
el mundo solo tendrá una generación de idiotas.*

Albert Einstein

2.1. Descripción del problema	3
2.2. Introducción	4
2.3. Objetivo	6
2.3.1. Objetivos específicos	6

2.1 DESCRIPCIÓN DEL PROBLEMA

Se tiene una serie de vehículos aéreos no tripulados (drones) y un área agrícola en la cual es necesario optimizar sus condiciones físico-químicas. Los drones se emplean tanto para realizar las tareas (riego, fertilización y detección de plagas) mediante el uso de sensores que les permiten interactuar con el ambiente. Los drones presentan ciertas limitaciones: capacidad limitada de energía, altura máxima de vuelo, rango de alcance, capacidad de carga, entre otras; por lo cual, se hace necesario encontrar las condiciones adecuadas para realizar las labores encomendadas.

2.2 INTRODUCCIÓN

Esta tesis aborda un problema llamado «Sistema multiagente para el manejo óptimo de un cultivo empleando la metodología Prometheus». La principal meta del trabajo propuesto es mediante el uso de drones, optimizar las condiciones de un cultivo cualquiera. Para ello, se ha diseñado un sistema multiagente, el cual permite simular unas condiciones iniciales al igual que el trabajo en el terreno. Los drones, tienen una capacidad limitada de energía, por lo cual deben recargar la batería en estaciones de carga (bases) cada vez que lleguen a un nivel crítico (volviendo a ellas desde el punto donde se encuentren) para así continuar con las labores hasta que se trate totalmente las zonas que así lo requieran.

Este problema se ha concebido dado el auge del uso de drones en actividades civiles, especialmente en aquellas de vital importancia para el hombre (como lo es la agricultura). Si bien, este no es un tema nuevo, ha tomado un particular interés durante los últimos años gracias a las facilidades para adquirir o diseñar un vehículo de estas características [1].

El uso de drones en las actividades agrícolas, puede traer consigo muchos beneficios para los productores [2]:

1. Sobrevolar el terreno de una forma rápida y captar información diversa gracias a los sensores que llevan instalados. Permitiendo gestionar las etapas de los cultivos, controlar e incrementar la productividad.
2. Un solo dron puede monitorizar cientos de hectáreas de forma precisa, evaluando las condiciones del terreno, con el fin de recoger información sobre el nivel de humedad, la temperatura o la tasa de crecimiento de los cultivos.
3. Localización temprana y tratamiento de plagas que arruinen parte de la cosecha.

4. Acceso a zonas difíciles (bosques, laderas, cultivos en etapa de madurez, etc.)

Todo esto representa un ahorro de recursos significativo para el agricultor [3]:

1. Reducción de la cantidad de productos químicos que se emplean en los cultivos, lo que disminuye el gasto de adquisición y la contaminación del subsuelo por el exceso de agroquímicos o el derrame innecesario del mismo.
2. Fumigación por medio de los drones sin contar con agentes externos, haciéndolo de una forma focalizada para aquellas zonas que lo requieran, evitando la afectación de áreas sanas.
3. Controlar el funcionamiento del riego, permitiendo la irrigación solo en las áreas que lo requieran, lo que traduce un ahorro en el agua empleada.

Ahora bien, la solución planteada consta de un conjunto de drones distribuidos en un terreno o cultivo, allí recorrerán el terreno midiendo con la ayuda de sensores las zonas (parcelas) que requieren fertilizantes, las que tienen mayor o menor humedad y aquellas que están siendo afectadas por alguna plaga (Capítulo 4). Se comunicarán en tiempo real entre ellos y con los agricultores para facilitar las tareas, obtener una mejor coordinación y resultados efectivos. Con el fin de facilitar el trabajo, la metodología se dividió en dos etapas:

1. La primer fase consta del diseño del sistema multiagente (ver Capítulo 5).
2. La segunda fase analiza el movimiento de los drones en el terreno a través de la construcción de técnicas de búsqueda heurística (ver Capítulo 6).

Para evaluar el desempeño de las soluciones, se planteó un diseño de experimentos (ver Capítulo 7), teniendo en cuenta las necesidades computacionales, restricciones del problema y supuestos planteados. Los resultados de la experimentación se pueden consultar en el capítulo 8. Al finalizar, se presentan algunas conclusiones y el trabajo que se pretende continuar (Capítulo 9).

2.3 OBJETIVO

Diseñar un sistema multiagente empleando la metodología Prometheus para el manejo eficiente de un cultivo.

2.3.1 OBJETIVOS ESPECÍFICOS

1. Optimizar la condiciones de un cultivo mediante el uso de drones.
2. Maximizar el número de zonas atendidas por los drones.
3. Minimizar el uso de la batería de los drones al moverse por el terreno y al realizar las actividades.
4. Automatizar el movimientos de los drones.
5. Maximizar el desplazamiento de los drones por el terreno.

CAPÍTULO 3

MARCO TEÓRICO

*El genio comienza las grandes obras,
pero sólo el trabajo las acaba.*

Joseph Joubert

3.1. Sistemas de agentes	7
3.2. Metodología Prometheus	10
3.3. Agricultura de precisión	11
3.3.1. ¿Qué es un vehículo aéreo no tripulado (VANT)?	12
3.3.2. Uso de drones en la agricultura	12
3.4. Heurísticas y metaheurísticas	13
3.4.1. Búsqueda local	13
3.4.2. Ascenso de colinas	14
3.4.3. Búsqueda Tabú	16

3.1 SISTEMAS DE AGENTES

Muchos sistemas construidos en la práctica (software) son de tipo “reactivo”;

es decir, mantienen una relación continua de interacción con su ambiente. Estos sistemas simplemente procesan una función con datos de entrada y terminan con la impresión de un resultado (un programa para calcular la suma de dos números, un procesador de texto, un optimizador, etc.).

Un agente es un sistema reactivo, cuya principal característica es la autonomía; definiéndose esta como la capacidad de que pueda actuar sin la intervención de un usuario. Según Weiss [4] “Un agente es un sistema computacional que está situado en un ambiente y que es capaz de tomar acciones autónomas en ese ambiente con el fin de cumplir sus objetivos de diseño”. Para Bordini [5], “un agente es un tipo de sistema reactivo, el cual tiene cierto grado de autonomía dada la delegación de tareas, pero toma sus propias decisiones con el fin de satisfacer sus objetivos”.

Los agentes, poseen un conjunto de propiedades [6], entre las cuales se tienen:

- Autonomía: Capacidad de decidir como operar independientemente para satisfacer un conjunto de objetivos.
- Proactividad: Capacidad de exhibir comportamiento orientado a objetivos; es decir, tomar la iniciativa para satisfacerlos.
- Reactividad: Capacidad de percibir su medio ambiente, y de responder de manera oportuna a cambios en el mismo para así satisfacer sus objetivos.
- Habilidad social: Capacidad de interactuar y cooperar con otros agentes (y humanos) sus conocimientos para satisfacer sus objetivos.
- Aprendizaje: Capacidad de aprender de sus observaciones y de experiencias pasadas para mejorar su desempeño.

De forma general, los agentes se ubican en algún ambiente (también llamado entorno); por lo cual, tienen la capacidad de percibirlo mediante sensores, tienen un posible repertorio de acciones y mecanismos de razonamiento para evaluarlas y

ejecutarlas mediante actuadores con el fin de modificar el entorno. De este proceso, se da la mejora a través de la experiencia. Cuando reúne todas estas características, se le llama “agente inteligente”.

La figura 3.1 representa el modelo general de agente descrito en el párrafo anterior.

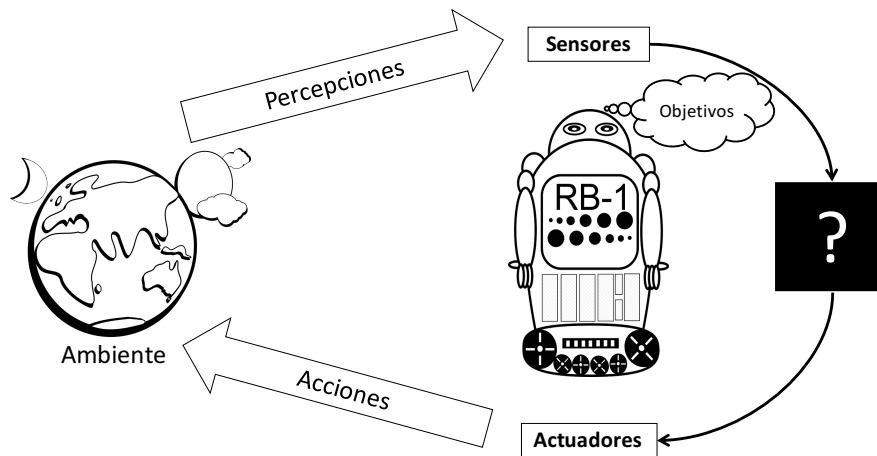


Figura 3.1: Modelo de agente

En la práctica, no es común ver trabajar a un solo agente; por lo general, se encuentran varios agentes cooperando entre si a través del intercambio de información (entorno social), así pueden satisfacer y alcanzar tanto sus metas individuales como los objetivos generales. En este caso, se habla de sistemas multiagentes.

Los sistemas multiagentes se ubican en un entorno que proporciona a cada participante información para modificar sus creencias, activando mecanismos de raciocinio que producen acciones y la comunicación con otros agentes. El ambiente donde interactúan contiene objetos, los cuales son percibidos, creados, destruidos o modificados, todo esto con un conjunto de operaciones propias de cada agente. En términos generales, para hablar de un sistema multiagente propiamente dicho, el sistema debe contener: un conjunto de agentes, un ambiente (entorno) y un conjunto de objetos [7].

Para el desarrollo de un sistema multiagentes, existen varias metodologías y cada una posee ciertas características según el proyecto a desarrollar. Entre las más conocidas se tiene la metodología Prometheus.

3.2 METODOLOGÍA PROMETHEUS

Es una metodología para desarrollar sistemas multiagentes mediante un proceso que consta de tres fases: especificación del sistema, diseño arquitectónico y diseño detallado [8].

La especificación del sistema consiste en identificar los actores y las interacciones de estos con el sistema mediante percepciones y acciones (entradas y salidas), desarrollar los escenarios que ilustran la operación de forma general (situaciones normales y sucesos alternativos), identificar los objetivos generales y los secundarios, identificar las fuentes y tipos de datos externos y definir el agrupamiento de los objetivos con los demás elementos para formar los roles básicos del sistema.

La fase de diseño arquitectónico determina los tipos de agentes con los elementos producidos en la fase anterior. Cada agente al menos, se hace responsable de un rol identificado anteriormente. De igual forma, se especifican las interacciones entre los agentes mediante los protocolos de interacción (estos incluyen los mensajes, percepciones y acciones) y muestra el diagrama general.

La fase de diseño detallado define los detalles internos de cada agente y especifica como estos realizarán sus tareas. Para ello, se especifican las capacidades que cada agente tiene, los eventos internos, los planes y la estructura de datos requeridas. Esta definición se realiza en un diagrama general de capacidad (estructura de los planes) y el comportamiento dinámico se describe con los diagramas de procesos.

Para el desarrollo de esta metodología se emplea la herramienta PDT (Prometheus Design Tool) (Padgham, Thangarajah, Paul) [9]. Esta herramienta permite

desarrollar el Sistema Multiagente siguiendo la metodología Prometheus.

Prometheus presenta algunos conceptos fundamentales:

- Percepción: información recolectada por algún sensor y que sirve como dato de entrada al sistema.
- Acción: es lo que hace los agentes para modificar su entorno. Esta puede ser interna o externa, simples y directas o interacciones complejas.
- Objetivo: para qué se construye el sistema y hacia dónde se dirige.
- Roles: describen funcionalidades que se cree debe tener el sistema, por regla general los roles están ligados a los objetivos.
- Escenarios: muestran una instancia particular del funcionamiento del sistema, consistiendo en una secuencia de pasos (acciones, percepciones y objetivos).

3.3 AGRICULTURA DE PRECISIÓN

La agricultura de precisión, consiste en el uso de herramientas tecnológicas para dar un manejo detallado y diferenciado a los cultivos a través del conocimiento de la variabilidad presente de cada parcela. Para lograr este cometido, se emplean GPS, imágenes satelitales y multiespectrales, sensores y demás herramientas que permitan interactuar con el entorno y medirlo de manera eficaz [10].

La agricultura de precisión remonta sus orígenes a los años ochenta, siendo de los países pioneros Japón con el uso de drones en las zonas rurales como medida para atraer a los jóvenes hacia el campo y tecnificar la producción agrícola.

Los beneficios de la utilización de estas herramientas son numerosos: permite reducir costos, mejorar la rentabilidad, optimizar la producción y disminuir el impac-

to ambiental al emplear los recursos de forma dirigida y ajustada a los requerimientos reales [11].

3.3.1 ¿QUÉ ES UN VEHÍCULO AÉREO NO TRIPULADO (VANT)?

Un vehículo aéreo no tripulado “VANT”, por sus siglas en español o UAV por las siglas en inglés de “Unmanned Aerial Vehicle” o simplemente drone es una aeronave que vuela sin tripulación a bordo y es controlado de forma remota [12].

En sus inicios, era destinado primordialmente al ámbito militar, principalmente en el reconocimiento del terreno y realizando ataques. No obstante, esto a cambiando con el tiempo y actualmente se dio apertura hacia el uso civil, destacando su uso en actividades agrícolas como vigilancia de los cultivos e irrigación del terreno.

3.3.2 USO DE DRONES EN LA AGRICULTURA

El uso de VANT en las actividades agrícolas presenta una serie de ventajas, dado su fácil uso y adquisición; por lo cual, facilita a los agricultores el acceso a información en tiempo real sobre el estado hídrico, crecimiento y salud de los cultivos mediante la obtención de información (empleando sensores infrarrojos, cámaras, radares e imágenes) y su posterior análisis [13].

Entre las múltiples posibilidades del uso de drones, se tienen [14]:

- Pueden sobrevolar los campos de una forma rápida y captar información diversa gracias a los sensores que llevan instalados. Esto permite gestionar las etapas de los cultivos, controlar e incrementar la productividad.
- Un solo dron puede monitorizar cientos de hectáreas de forma precisa, evaluando las condiciones del terreno, con el fin de recoger información sobre el

nivel de humedad, la temperatura o el ritmo de crecimiento de los cultivos.

- Localización prematura de plagas que arruinen parte de la cosecha.
- Acceso a zonas difíciles (bosques, laderas, cultivos avanzados, etc.).

3.4 HEURÍSTICAS Y METAHEURÍSTICAS

Se habla de heurística para referirse a una técnica, método o procedimiento inteligente de realizar una tarea que no es producto de un riguroso análisis formal, sino de conocimiento experto sobre la tarea. En especial, se usa el termino heurístico para referirse a un procedimiento que trata de aportar soluciones a un problema con un buen rendimiento, en lo referente a la calidad de las soluciones y a los recursos empleados [15].

En muchas ocasiones, una heurística no es eficiente por si sola, requiere de estrategias generales que la guíen a encontrar mejores soluciones. En este punto se habla de “metaheurísticas”, las cuales combinan de forma inteligente diversas técnicas para explorar el espacio de soluciones. Como definen Osman y Kelly [16]: “Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son ni efectivos ni eficientes. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y la mecánica estadística”.

3.4.1 BÚSQUEDA LOCAL

La búsqueda local está basada en el mejoramiento paso a paso de una función de costo al explorar las vecindades de soluciones cercanas. El uso del algoritmo de

búsqueda local presupone la definición de una solución, una función de costo y una estructura de vecindades [17]:

- La definición de una solución y su función de costo, se establece al momento de establecer si el problema busca maximizar o minimizar un objetivo principal.
- La estructura de vecindades son todas las posibles soluciones que se consideran en cada punto evaluado.

La ventaja de usar una búsqueda local, radica en la posibilidad de encontrar soluciones muy rápidamente, pero dado este aspecto su principal desventaja es que queda atrapada fácilmente en mínimos locales.

El algoritmo 1 presenta el pseudocódigo base de una búsqueda local, donde se inicia con un solución inicial x , se determinan los vecinos $V(x)$ y se elige uno de ellos (x') si mejora la solución actual:

Algoritmo 1 Búsqueda local general

```

1:  $x = 0$ 
2: mientras  $x$  no es óptimo local hacer
3:   Determinar  $V(x)$ 
4:   Elegir  $x' \in V(x)$  si mejora solución actual
5:   Hacer  $x = x'$ 
6: fin mientras

```

3.4.2 ASCENSO DE COLINAS

Ascenso de colinas (Hill Climbing) es un algoritmo de búsqueda local. Es lo que se conoce como algoritmo voraz, pues siempre hace la mejor elección disponible en cada paso, con la esperanza de que de esta manera se pueda obtener el mejor resultado global. La solución actual se transforma mediante alguna operación que puede ser una mutación, y si la operación proporciona una solución con mayor aptitud que

la anterior, se conserva la nueva; en caso contrario, se conserva la solución actual. Si el nuevo punto del espacio de búsqueda es mejor, se transforma en el punto actual, sino otro punto vecino es seleccionado y evaluado. El método termina cuando no hay mejoras, o cuando se alcanza un número predefinido de iteraciones [18].

El algoritmo inicia con una solución inicial x , la cual corresponde al punto actual x_0 , a partir de el se determinan los vecinos $V(x)$ y se guardan en una lista L ; de igual manera, se crean las variables *ProxEval* y *ProxNodo* las cuales contendrán el valor de la función objetivo y el punto siguiente respectivamente. Cada x_L (vecinos de la solución actual) es evaluado ($Eval(x_L)$), si este mejora la solución actual, se escoje como el siguiente movimiento y se guarda la solución generada; en caso contrario, se continua en el mismo punto y se evalúa el siguiente vecino.

El algoritmo 2 presenta el pseudocódigo para la heurística descrita anteriormente:

Algoritmo 2 Búsqueda Hill Climbing general

```

1:  $x_1 = x_0$ 
2: mientras no haya solución hacer
3:    $L = V(x)$ 
4:    $ProxEval = -INF$ 
5:    $ProxNodo = NULL$ 
6:   para todo  $x$  en  $L$  hacer
7:     si  $Eval(x_L) > ProxEval$  entonces
8:        $ProxNodo = x_L$ 
9:        $ProxEval = Eval(x)$ 
10:    si no, si  $ProxEval \leq Eval(x)$  entonces
11:      devolver  $x$ 
12:    fin si
13:  fin para
14: fin mientras

```

3.4.3 BÚSQUEDA TABÚ

La búsqueda tabú es una metaheurística que busca dar “inteligencia” a los algoritmos de búsqueda local. Fue introducida por Glover [19] y se caracteriza por utilizar una estrategia basada en el uso de estructuras de memoria para escapar de los óptimos locales en los que se puede caer al moverse de una solución a otra en el espacio de soluciones [20].

La búsqueda tabú presenta como elementos [21]:

- El uso de estructuras de memoria la cual puede ser de corto plazo (memoria reciente) y de largo plazo (memoria de frecuencias).
- La lista tabú y los mecanismos de selección del siguiente movimiento.
- Las estrategias de búsqueda: intensificación y diversificación.

La memoria a corto plazo (reciente), almacena los últimos movimientos por un lapso denominado tiempo de permanencia (tabu tenure), esta memoria se representa en la lista tabú.

La lista tabú almacena las soluciones o atributos (componentes de soluciones) que no deben ser elegidos por un tiempo determinado. No obstante, elementos de esta lista pueden ser elegidos mediante un criterio de aspiración, el cual marca al atributo como admisible aún estando en esta lista [22].

La memoria a largo plazo favorece la búsqueda en regiones donde se han encontrado buenas soluciones. Este proceso se conoce como de intensificación y se lleva a cabo dándoles alta prioridad a soluciones similares a la actual y penalizando las que se alejen de ella. Una vez terminado el proceso de intensificación puede resultar conveniente explorar regiones nuevas en el espacio de búsqueda, dándose un proceso de diversificación para favorecer soluciones con características diferentes a la solución actual [23].

3.4.3.1 METODOLOGÍA BÚSQUEDA TABÚ

La búsqueda tabú actúa de la siguiente manera: dada una solución x se define un vecindario $V(x)$, se genera la lista tabú $T(x, k)$, una lista $A(s, k)$ que conforma un conjunto de aspirantes y un vecindario reducido $V^*(x)$, el cual se evalúa y se elige la mejor solución x que mejore la actual. Finalmente, se actualiza la lista tabú $T(x, k)$ y se continúa iterando hasta que no se halle una mejor solución.

El algoritmo 3, muestra el pseudocódigo de la búsqueda tabú simple, la cual se empleó en este trabajo:

Algoritmo 3 Búsqueda Tabú simple general

```

1:  $k = 1$ 
2:  $x = x_0$ 
3: mientras no haya solución hacer
4:   Determinar  $V(x)$ 
5:   Determinar  $T(x, k)$ 
6:   Determinar  $A(s, k)$ 
7:   Determinar  $V^*(x, k) = V(x, k) - T(x, k) \cup A(s, k)$ 
8:   Elegir la mejor  $x \in V^*(x)$  si mejora solución actual
9:   Actualizar  $T(x, k)$ 
10:   $K = k + 1$ 
11: fin mientras

```

Para aplicar la búsqueda tabú, es necesario estar familiarizado con el problema a tratar, conocer su naturaleza, la forma de las soluciones factibles de tal manera que pueda sugerir la configuración de la lista tabú, del vecindario y el criterio de aspiración.

CAPÍTULO 4

MODELACIÓN DEL PROBLEMA

*La vocación del político de carrera
es hacer de cada solución un problema.*

Woody Allen

4.1. Entrada de datos	19
4.2. Salida de datos	19
4.3. Función objetivo	20
4.4. Restricciones	20

La modelación del problema se divide en dos secciones. La primera sección muestra los elementos que componen la entrada de datos necesarios para dar inicio al proceso; la segunda sección muestra la salida que arroja el ejercicio y las restricciones que se deben tener en cuenta para considerar la solución como factible.

Indices de arreglos

- size.* Dimensiones de la cuadrícula.
- i.* Cantidad de bases.
- j.* Cantidad de zonas a trabajar.
- k.* Cantidad de drones.
- level.* Nivel de batería.

x, y . Coordenada en la cuadrícula.
 $step$. Cantidad de movimientos disponibles.

Parámetros

$battery_{level}$. Capacidad de carga.
 $agPos_{x,y}$. Posición del dron.
 $total_{step}$. Total de movimientos disponibles.
 C_j . Costo del movimiento.

Parámetros de preprocesamiento

$d_{agPos,base}$. Distancia manhattan entre el punto $agPos_{x,y}$ y $base_i$.

4.1 ENTRADA DE DATOS

El área a trabajar se representa por una cuadrícula de tamaño $Grid_{size}$, la cual contiene un conjunto de bases de carga $base_i$, un conjunto de puntos vacíos $zero_j$ y un conjunto de zonas a trabajar $garb_j$. Cada $garb_j$ representa una condición a trabajar (riego, fertilización, fumigación).

El conjunto de drones se denota por $drone_k$. Cada $drone_k$ presenta una capacidad limitada de carga $battery_{level}$, una posición en la cuadrícula $agPos_{x,y}$ y un total de movimientos disponibles $total_{step}$ (pueden ser menores o iguales al total de la carga que posea el dron).

4.2 SALIDA DE DATOS

La solución se compone de un conjunto de movimientos para cada $drone_k$ a través del $Grid_{size}$ atendiendo los diferentes puntos $Garb_j$ con la batería dada $battery_{level}$ y el total de movimientos disponibles $total_{step}$.

4.3 FUNCIÓN OBJETIVO

La función objetivo de este problema es atender el mayor número de zonas con el menor costo posible:

$$\sum_{j \in J} (C_j * total_{step}) drone_k.$$

El costo de visitar un punto vacío es de una unidad y el de visitar una posición con necesidad de tratamiento es de cero.

4.4 RESTRICCIONES

La solución a este problema debe tener en cuenta las siguientes restricciones.

El total de pasos disponibles para cada dron no puede exceder la duración de la batería:

$$total_{step} \leq battery_{level}.$$

Cada punto que requiera tratamiento (riego, fertilización, fumigación) debe ser visitado:

$$\sum_{j \in J} Garb_j = 1.$$

Un punto no debe ser visitado dos veces:

$$\sum_{j \in J} Garb_j + \sum_{x \in X} \sum_{y \in Y} zero_{x,y} < 2.$$

CAPÍTULO 5

DISEÑO DEL SISTEMA MULTIAGENTE

*Dondequiera que se implanten muchas máquinas para reemplazar a los hombres,
encontraremos muchos hombres que no serán otra cosa que máquinas.*

Louis de Bonald

5.1. Modelado del sistema	22
5.1.1. Presentación del problema	22
5.1.2. Fase especificación del sistema	23
5.1.3. Fase diseño arquitectónico	28
5.1.4. Fase diseño detallado	31

PARA el diseño del sistema multiagente, se utilizó la metodología Prometheus, la cual se trabajó en tres fases:

- Especificación del sistema: identifica las funciones básicas.
- Diseño arquitectónico: determina los agentes y las interacciones.
- Diseño detallado: describe los elementos internos de los agentes.

5.1 MODELADO DEL SISTEMA

5.1.1 PRESENTACIÓN DEL PROBLEMA

La descripción del problema se mencionó en la sección 2.1. Este problema ha sido acotado con el fin de dar una mejor perspectiva desde el enfoque multiagente y así facilitar su estudio. Como punto de partida, se han planteado las siguientes suposiciones:

- Se cuenta con el mapa del terreno y la localización de las zonas a optimizar es conocida.
- El recorrido debe tener en cuenta las diferentes condiciones del suelo, es decir, en una misma área pueden existir diferentes necesidades según la condición a evaluar: humedad, fertilizantes, pesticidas, cultivos, etc.
- El tiempo de vuelo no puede exceder la capacidad de la batería.
- Cada punto con necesidad de trabajo debe ser visitado al menos una vez.
- Los drones solo pueden recargar la batería en las estaciones dedicadas.
- Un dron puede sobrevolar por un punto dos veces, siempre y cuando lo haga para recargar la batería.
- Al final del ejercicio, todos los drones deben estar en las estaciones de carga.

Para facilitar la comprensión del documento, se presenta la simbología que compone el sistema (figura 5.1):

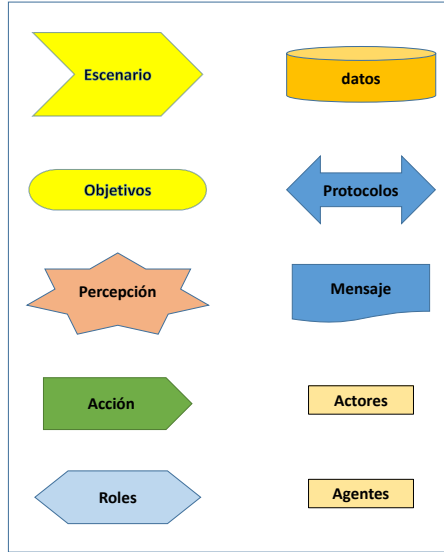


Figura 5.1: Simbología del sistema

5.1.2 FASE ESPECIFICACIÓN DEL SISTEMA

5.1.2.1 INTERACCIÓN DEL SISTEMA CON EL ENTORNO

La primera fase inicia con un diagrama (*Analysis Overview*) para mostrar de forma preliminar las interacciones presentes entre el sistema y su ambiente (figura 5.2). Dentro de este diseño, se han identificado cuatro actores:

- Base (*base*).
- Dron irrigador (*drone1*).
- Dron fertilizador (*drone2*).
- Dron fumigador (*drone3*).

El actor Base representa el sitio donde se recargan los drones. Este proceso se describe en el escenario *Recargar Escenario* mediante las acciones *recargar(batería)*, *moverHacia(base)* y *parar*; las percepciones *batería(nivel)*, *base(x,y)* y *pasos*.

Los actores *drone1*, *drone2* y *drone3* representan a los VANT encargados de suministrar el agua tanto al terreno como al cultivo, los agroquímicos y de controlar las plagas detectadas respectivamente según lo descrito en el escenario *Optimizar Escenario*, a través de las acciones *optimizar(riego)*, *optimizar(abono)*, *optimizar(fumigar)* (las cuales se agrupan para mejor manejo en *optimizar(zona)*), *mover(zona)* y *moverAleatorio*; las percepciones *pos(x,y)*, *agente(zona)*, *tamañoCuadrícula(x,y)* y *requiereTratamiento(zona)*.

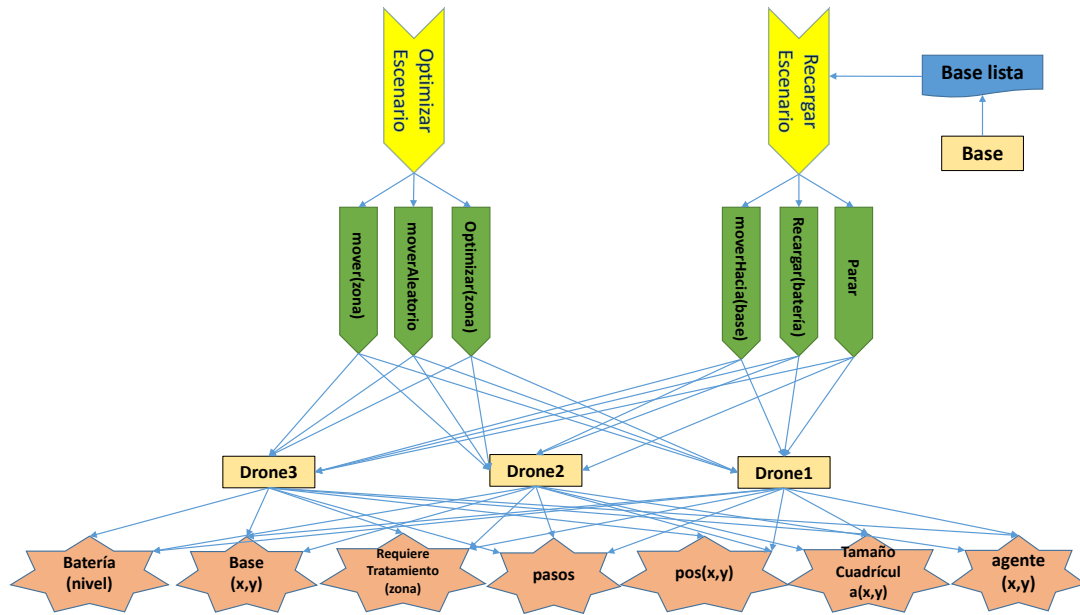


Figura 5.2: Análisis preliminar

5.1.2.2 ESCENARIOS

Los escenarios describen los pasos que se realizan en alguna situación que se presente en el sistema; estos se detallan en el diagrama *Scenario Overview* (figura 5.3). Los escenarios planteados son: *Recargar Escenario* (correspondiente a la fase de recarga de batería) y *Optimizar Escenario* (presenta la realización de las tareas de optimización tanto del terreno como de los cultivos).

Los pasos que describen cada escenario se presentan a continuación:

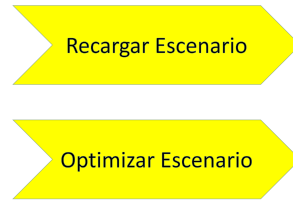


Figura 5.3: Escenarios

- *Recargar Escenario* inicia con la percepción $base(x,y)$, la cual le indica al dron la posición donde se encuentra la base más cercana para la recarga y la percepción $batería(nivel)$ le indica el nivel de batería disponible. Si el nivel se encuentra bajo, se activa la acción $moverHacia(base)$, por la cual se dirige a la base de carga, teniendo en cuenta la percepción $agente(zona)$ que le indica si hay un dron en el camino para evitar una colisión. Al llegar a la base, se activa la acción $recargar(batería)$ y al finalizar retorna a su trabajo. Este proceso se muestra en el diagrama 5.4:

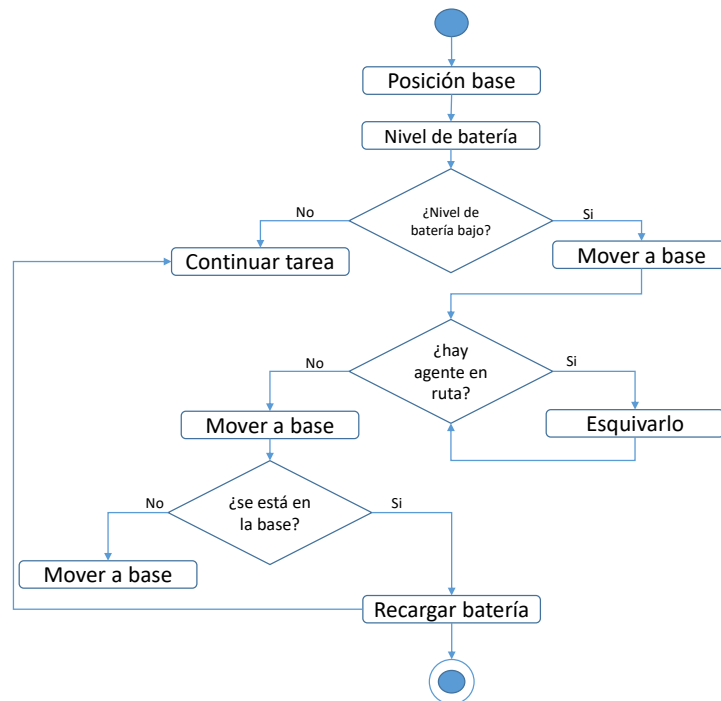


Figura 5.4: Diagrama recargar escenario

- *Optimize Scenario* comprende las tareas para controlar el nivel de humedad, suplir las necesidades de agroquímicos y suministrar plaguicidas al terreno y/o el cultivo. Cada dron está acondicionado para realizar una tarea específica por lo cual, la percepción *needstreatment(zone)* se convierte en *needsIrrigate(zone)* para el drone1, *needsFertilize(zone)* para el drone2 y *needsFumigate(zone)* para el drone3; así, cada dron trabaja un área específica y se atienden zonas con diferentes necesidades al mismo tiempo.

Para trabajar cada zona, el dron revisa a su alrededor si alguna zona requiere trabajo, de ser así se desplaza a ese punto con la acción *mover(zona)* donde “zona” indica un punto cardinal (norte, sur, este y oeste) y trabaja en ese punto con la acción *optimizar(zona)* (la cual varía según el tipo de dron en *optimizar(riego)*, *optimizar(abono)* y *optimizar(fumigar)*). En caso de no encontrar un punto con necesidad de tratamiento, se mueve hacia otro punto con la acción *moverAleatorio*, la cual activa el algoritmo de búsqueda para obtener el mejor movimiento. En cada movimiento, se usa la percepción *agente(x,y)*, de esta forma el dron revisa si la posición a donde se moverá no la ocupa otro dron; también revisa el nivel de batería con la percepción *batería(nivel)* para asegurarse de contar con la carga necesaria y la percepción *pasos* que le indica el total de pasos disponibles para avanzar, los cuales son fijados al inicio del ejercicio. Este proceso se muestra en el diagrama 5.5:

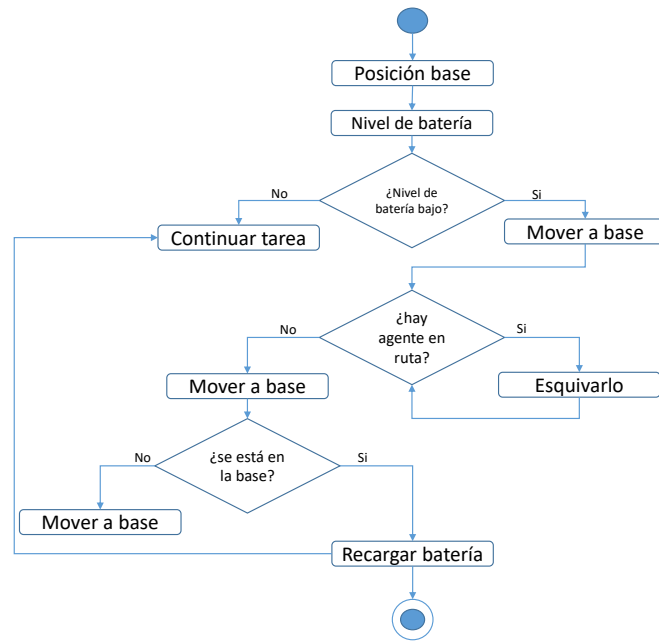


Figura 5.5: Diagrama optimizar escenario

5.1.2.3 OBJETIVOS DEL SISTEMA

En el diagrama *Goal Overview* (figura 5.6) se muestran los objetivos planteados; estos a su vez se descomponen en subobjetivos (establecen relación hijo-padre) describiendo como se alcanza el objetivo.

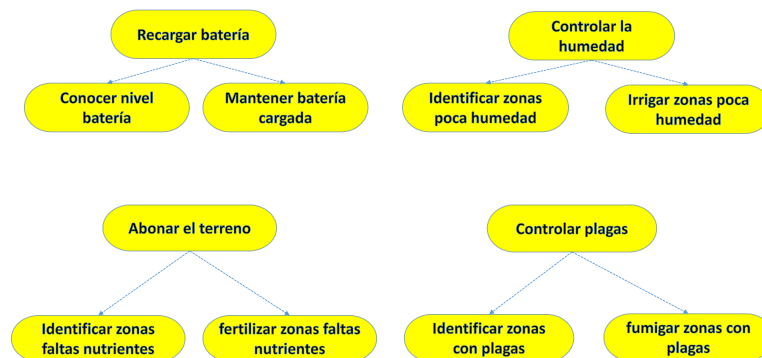


Figura 5.6: Objetivos

5.1.2.4 ROLES DEL SISTEMA

Los roles son obtenidos mediante la agrupación de objetivos y subobjetivos con la relación pertinente de acciones y percepciones (figura 5.7). Dadas las agrupaciones efectuadas, se plantearon dos roles para el sistema:

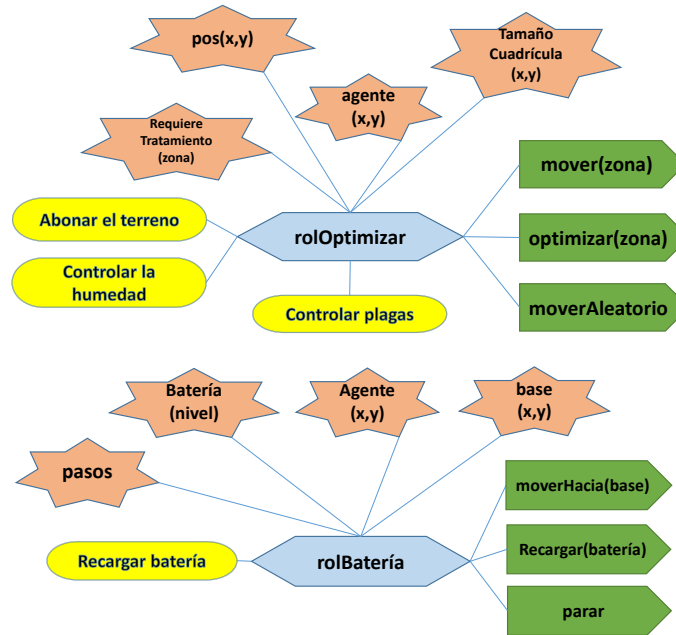


Figura 5.7: Roles del sistema

5.1.3 FASE DISEÑO ARQUITECTÓNICO

5.1.3.1 IDENTIFICACIÓN DE LOS AGENTES

Para identificar los agentes, se agruparon los roles de forma tal que cada actor esté a cargo de uno o más roles. El resultado de este ejercicio se describe en la figura 5.8.

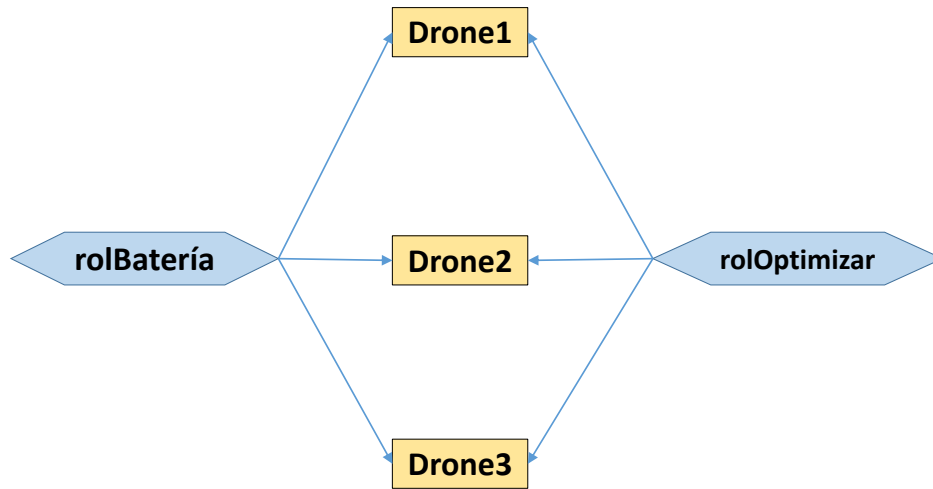


Figura 5.8: Definición de agentes

5.1.3.2 ACOPLAMIENTO DE DATOS

Para que exista la correcta funcionalidad del sistema, es necesario contar con ciertos datos asociados a roles. Esta relación se muestra en el diagrama *Data Coupling Overview* (figura 5.9). El rol *Batería* tiene como objetivo recargar la batería las veces que sea necesario. Para ello, los agentes acceden al dato *ubicación bases* y producen el dato *cantidad recargas*. El rol *Optimizar* tiene por objetivo optimizar las condiciones del terreno; para satisfacerlo, sus agentes asociados acceden al dato *tamaño del terreno*, para conocer las posiciones de los sitios donde deben realizar las tareas y producen los datos *posiciones optimizadas* y *lista restringida de posiciones*.

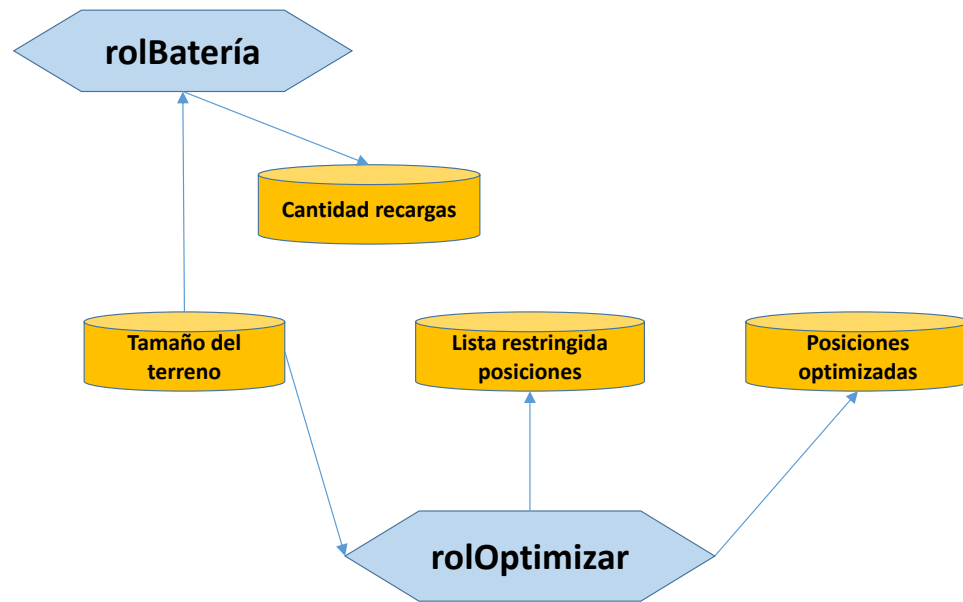


Figura 5.9: Acoplamiento de datos

5.1.3.3 INTERACCIONES ENTRE AGENTES

En este apartado se describen las relaciones entre los agentes (figura 5.10). Se han planteado los protocolos de interacción para observar la relación existente entre los agentes que componen el sistema y el entorno en el cual se encuentran.

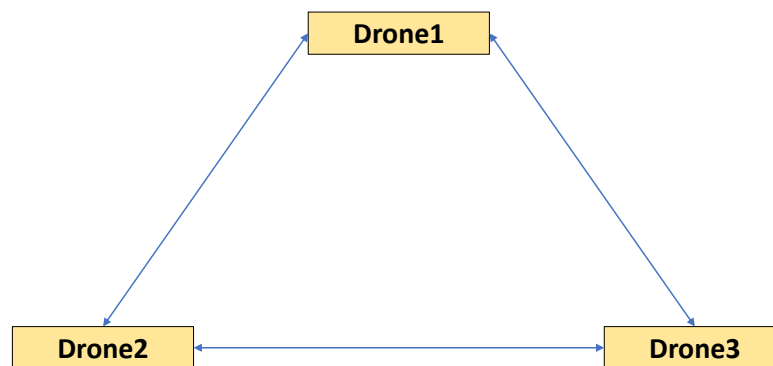


Figura 5.10: Interacciones entre agentes

5.1.3.4 ARQUITECTURA GLOBAL DEL SISTEMA

El diagrama System Overview (figura 5.11) muestra la arquitectura global del sistema (agentes, percepciones, acciones, datos y protocolos de interacción) en una única vista, incluyendo las relaciones de comunicación entre los agentes.

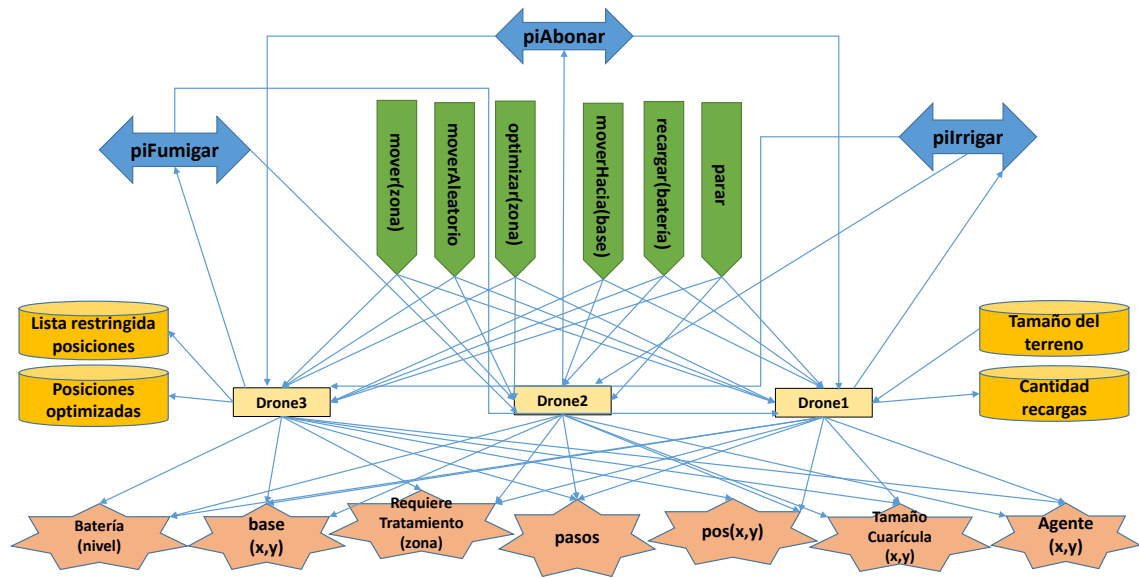


Figura 5.11: Arquitectura global

5.1.4 FASE DISEÑO DETALLADO

En esta fase se desarrollan los elementos internos de los agentes, en función de capacidades, datos, eventos y planes. Este diseño se muestra en la figura 5.12.

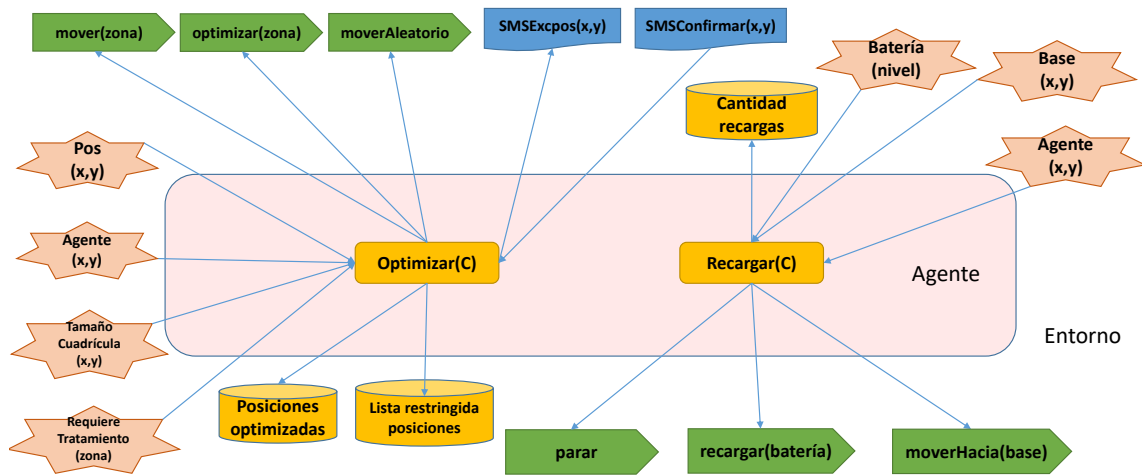


Figura 5.12: Capacidad agentes

Las tablas que contienen los descriptores del diseño se encuentran en el apéndice A. Allí se ordenaron de acuerdo a las tres fases de la metodología utilizada, explicando de forma detallada los componentes presentados en este capítulo.

CAPÍTULO 6

ALGORITMOS IMPLEMENTADOS

*Aunque los caminos de la búsqueda son numerosos,
la búsqueda es siempre la misma.*

Rumi

6.1. Búsqueda local	33
6.2. Búsqueda ascenso de colinas (Hill Climbing)	34
6.3. Búsqueda Tabú	36

UNO de los aspectos importantes que se consideraron al desarrollar el modelo multiagente, fue el desplazamiento que hacen los drones en el terreno, el cual debe consumir poca batería y permitir realizar las labores encomendadas. Para este fin, se diseñaron tres algoritmos, los cuales se emplean en la simulación.

6.1 BÚSQUEDA LOCAL

Este algoritmo funciona de la siguiente forma:

1. El drone se ubica en una zona de la cuadrícula (solución inicial x).

2. A partir de esa posición se generan cuatro vecinos $V(x)$ (izquierda, derecha, arriba y abajo).
3. Este vecindario se evalúa y con una distribución aleatoria uniforme se elige alguno de esos vecinos (teniendo en cuenta que la posición a elegir no pueda estar ocupada por otro drone).
4. Al finalizar, el drone se mueve hacia la posición escogida x .

El algoritmo 4 ilustra la búsqueda local descrita en el párrafo anterior:

Algoritmo 4 Búsqueda local

```

1:  $k = 0$ 
2:  $x = x_0$ 
3: mientras no se cumpla criterio de parada hacer
4:   Determinar  $V(x)$ 
5:   Elegir  $x_k$  de forma aleatoria
6:   si posición  $x_k$  está ocupada entonces
7:     Elegir otro  $x_k$  de forma aleatoria
8:   fin si
9:   Hacer  $x = x_k$ 
10:   $K = k + 1$ 
11: fin mientras
  
```

6.2 BÚSQUEDA ASCENSO DE COLINAS (HILL CLIMBING)

La búsqueda de ascenso de colinas se trabajó ya que, dado su bajo costo computacional, permite encontrar soluciones de forma rápida, pero con el inconveniente de quedar atrapada en óptimos locales. El funcionamiento de este algoritmo se muestra a continuación:

1. A partir de una solución inicial x (posición del dron en el terreno), se genera una lista de vecinos $V(x)$ correspondiente a las posiciones adyacentes.
2. Se genera una lista $A(x)$, la cual contiene los vecinos que no estén ocupados por otro dron.
3. Se determina el conjunto $V^*(x)$ a partir de las listas $V(x)$ y $A(x)$, el cual contiene los vecinos disponibles y que no tienen un dron en esa posición.
4. La mejor solución x se escoge al revisar en el conjunto reducido de candidatos $V^*(x)$ las posiciones que tengan necesidad de tratamiento; si alguna de ellas requiere ser tratada, esta se escogerá, en caso contrario, si las posiciones están "vacías" (no requieren de trabajo) la selección se hará de forma aleatoria.

El proceso se ilustra en el algoritmo 5:

Algoritmo 5 Búsqueda Hill Climbing

- 1: $k = 0$
 - 2: $x = x_0$
 - 3: **mientras** no se cumpla criterio de parada **hacer**
 - 4: Determinar $V(x)$
 - 5: Determinar $A(x, k)$
 - 6: Determinar $V^*(x, k) = V(x) - A(x)$
 - 7: **si** $V^*(x, k)$ tiene vecinos empatados **entonces**
 - 8: Elegir x_k de forma aleatoria
 - 9: **si no**
 - 10: Elegir la mejor $x_k \in V^*(x)$
 - 11: **fin si**
 - 12: Hacer $x = x_k$
 - 13: $K = k + 1$
 - 14: **fin mientras**
-

6.3 BÚSQUEDA TABÚ

Este algoritmo se implementó con el fin de aprovechar las estructuras de memoria, para tener una búsqueda más “inteligente”. De esta forma, se busca escapar de los óptimos locales producidos por una simple búsqueda local. La búsqueda tabú desarrollada es la “búsqueda tabú simple”, ya que se adapta a las necesidades del problema. A continuación, se muestra la estructura planteada:

1. Se tiene una solución inicial x a partir de la posición del dron en la cuadrícula.
2. A partir de x se generan los vecinos, conformados por las posiciones izquierda, derecha, arriba y abajo. Estos se guardan en una lista $V(x)$.
3. Se genera una lista tabú, la cual contiene las posiciones visitadas con anterioridad (atributos) y que no pueden ser visitadas nuevamente (al inicio del algoritmo, la lista tabú estará vacía y se llenará con la posición actual), esta lista se denomina $T(x)$.
4. Se crea una tercera lista $A(x)$, la cual contiene las posiciones que no han sido visitadas y se encuentran en la lista $T(x)$, conformándose un primer conjunto reducido de candidatos.
5. El segundo conjunto reducido de candidatos, lo conforman las posiciones de $A(x)$ que no tienen un dron en ella, es decir, están libres. Estos candidatos se guardan en la lista $A'(x, k)$.
6. La mejor solución x se escoge al revisar en el segundo conjunto reducido de candidatos $V^*(x)$ las posiciones que tengan necesidad de tratamiento; si alguna de ellas requiere ser tratada, esta se escogerá, en caso contrario si las posiciones están “vacías”(no requieren de trabajo) la selección se hará de forma aleatoria.
7. La lista tabú se actualiza de dos formas: la primera guardando las posiciones que el dron haya visitado anteriormente y la segunda, a través de la comunica-

ción que tienen los drones: se enviarán entre si las posiciones que ya visitaron, para de esta forma prevenir a un “compañero” de visitar una posición en la que un tercero ya estuvo con anterioridad.

8. Se establece como criterio de aspiración (un atributo deja de ser tabú) cuando el dron tenga dos posibles movimientos: uno libre pero marcado como tabú y otro ocupado por un dron, en este momento, la posición libre deja de ser tabú para que el dron pueda moverse. También, al momento de no tener más movimientos disponibles, la lista tabú se reinicia y permite al dron moverse y construirla nuevamente. Así, puede recorrer su cuadrante para continuar con el trabajo.

El algoritmo 6 muestra el proceso anterior:

Algoritmo 6 Búsqueda Tabú Simple

```

1:  $k = 0$ 
2:  $x = x_0$ 
3: mientras no se cumpla criterio de parada hacer
4:   Determinar  $V(x)$ 
5:   Determinar  $T(x, k)$ 
6:   Determinar  $A(x, k) = V(x) - T(x)$ 
7:   Determinar  $A'(x, k)$ 
8:   Determinar  $V^*(x, k) = A(x, k) - A'(x)$ 
9:   si  $V^*(x, k)$  tiene vecinos empatados entonces
10:     Elegir  $x_k$  de forma aleatoria
11:   si no
12:     Elegir la mejor  $x_k \in V^*(x)$ 
13:   fin si
14:   Guardar  $x_k$  si mejora la solución actual  $x = x_k$ 
15:   Actualizar  $T(x, k)$ 
16:    $K = k + 1$ 
17: fin mientras

```

CAPÍTULO 7

MÉTODOLOGIA

*Busca dentro de ti la solución de todos los problemas,
hasta aquellos que creas más exteriores y materiales.*

Amado Nervo

7.1. Equipo computacional	39
7.2. Implementación computacional	39
7.3. Supuestos considerados	39
7.4. Modelos planteados	40
7.4.1. Primer modelo	41
7.4.2. Segundo modelo	41
7.4.3. Tercer modelo	42
7.5. Diseño de experimentos	42

A continuación se describen los diferentes supuestos planteados para desarrollar el trabajo de investigación, así como las herramientas computacionales usadas para resolver el problema multiagente.

7.1 EQUIPO COMPUTACIONAL

Los experimentos computacionales se realizaron en una iMac de 21.5 pulgadas con un procesador Intel Core i7 quad core de 3.3 GHz, memoria RAM de 16 GB, tarjeta gráfica Intel Iris Pro Graphics 6200 y Disco duro de 1 TB.

7.2 IMPLEMENTACIÓN COMPUTACIONAL

Para la implementación del sistema multiagente, se usó Jason versión 1.4.2 como interprete de AgentSpeak, el cual se instaló como un plug-in en el entorno de desarrollo Eclipse Java EE IDE for Web Developers versión Mars.2 Release (4.5.2), en lenguaje de programación Java.

7.3 SUPUESTOS CONSIDERADOS

Para el desarrollo de la simulación, se plantearon los siguientes supuestos:

1. Hay un agente para cada tarea: irrigar, fertilizar y abonar.
2. El terreno no presenta obstáculos (salvo los mismos agentes).
3. Los espacios en la cuadrícula (zonas) tienen iguales dimensiones y no son a escala.
4. Los drones cuentan con una batería limitada, por lo cual pueden “apagarse” si el nivel de batería es bajo.
5. La recarga de la batería la realiza cada drone en la base asignada.

7.4 MODELOS PLANTEADOS

Para analizar el problema planteado desde varias perspectivas, se plantearon tres modelos de simulación. Estos, comparten una estructura de diseño similar en cuanto a las condiciones iniciales; la diferencia entre ellos radica en el algoritmo usado para desplazarse por la cuadrícula.

Los pasos comunes a los modelos se listan a continuación:

1. Se construyó una cuadrícula de dimensiones $n \times n$ (con $n = 20$ y 30).
2. Se consideraron las tres condiciones del terreno (irrigar, fertilizar y abonar), donde cada una ocupará un número determinado de zonas.
3. La cuadrícula se llena según las condiciones establecidas al inicio siguiendo una distribución aleatoria uniforme, para así simular una imagen multiespectral, identificando cada zona a que condición pertenece.
4. Se distribuyeron cuatro grupos de agentes, cada uno con los tres agentes descritos (drone1, drone2 y drone3) ubicados en 4 bases de carga.
5. La cuadrícula se divide en cuadrantes y se ubica un grupo por cuadrante.
6. Los grupos se ubican en las esquinas de la cuadrícula, al lado de cada base.
7. Las bases se ubican en las esquinas del terreno.
8. Los agentes se mueven por su respectivo cuadrante, revisando el terreno y según sus capacidades, objetivos y planes (ver sección 5.1.4), realizan las tareas de riego, fertilización o fumigación.
9. Los agentes se reconocen entre si, por lo cual al encontrarse en el camino, se esquivan para evitar colisiones.
10. El criterio de parada se estableció para un número determinado de movimientos o bien cuando los drones se queden sin batería.

11. Se estableció el número máximo de pasos igual a la duración de la batería, de forma tal que se alcanzé el máximo trabajo posible con una carga completa de energía.

Las diferencias entre los modelos se listarán en los numerales siguientes.

7.4.1 PRIMER MODELO

En esta sección, se describe las diferencias a resaltar del primer modelo:

1. El movimiento que hacen los agentes por la cuadrícula lo realizan con una búsqueda local (ver sección 6.1), eligiendo de su vecindario una posición de forma aleatoria uniforme, teniendo en cuenta que otro dron no debe estar en la posición elegida.
2. Los drones recargan la batería cada vez que llegan a un valor crítico (15 %), para lo cual se dirigen a la base, allí renuevan el nivel de batería y regresan al trabajo.
3. Los drones no presentan comunicación entre ellos.

7.4.2 SEGUNDO MODELO

Las características del segundo modelo son:

1. Los agentes hacen el movimiento por la cuadrícula empleando un algoritmo de búsqueda local (hill climbing). La selección del movimiento se hace eligiendo la mejor posición o si se presenta empate mediante una distribución aleatoria uniforme (ver sección 6.2).

2. Los drones recargan la batería cada vez que llegan a un valor crítico (15 %), para lo cual se dirigen a la base, allí renuevan el nivel de batería y regresan al trabajo.
3. Los drones no presentan comunicación entre ellos.

7.4.3 TERCER MODELO

El tercer modelo tiene en cuenta:

1. Los agentes hacen el movimiento por la cuadrícula empleando una búsqueda tabú (ver sección 6.3), eligiendo la mejor posición posible.
2. No se le permite a los agentes pasar dos veces por la misma posición (salvo que lo haga para recargar batería).
3. Los drones recargan la batería cada vez que llegan a un valor crítico, el cual es calculado a partir de la distancia Manhattan [24] entre el agente y la base. Si esta distancia es menor o igual a la capacidad de carga del drone, este debe dirigirse a la base, recargar y retornar al trabajo.
4. Los agentes presentan comunicación entre ellos: transmiten a sus compañeros de cuadrantes las posiciones donde estuvieron, para así evitar que otro la visite nuevamente.

7.5 DISEÑO DE EXPERIMENTOS

Para los tres modelos propuestos se usó un diseño factorial completo aleatorizado 2^3 con dos niveles y tres variable de control, con 10 réplicas por tratamiento, generando 60 diferentes tratamientos.

CAPÍTULO 8

EXPERIMENTACIÓN

Vivir es conocer y experimentar esta vida humana.

Acertar o no es secundario.

Juan José Benítez

8.1. Primer modelo	43
8.2. Segundo modelo	46
8.3. Tercer modelo	48
8.4. Análisis general	50

En este capítulo se mostrarán los resultados obtenidos en la simulación del sistema multiagente según las condiciones planteadas.

8.1 PRIMER MODELO

Se realizaron 10 réplicas para cada tamaño de instancia (20 y 30), de forma tal que se pudiera evaluar el desempeño de los drones en el terreno.

- **Instancia igual a 20**

La gráfica 8.1 muestra el costo obtenido al trabajar en el terreno, donde se tiene un valor mínimo de 1082 y un valor máximo de 1095:

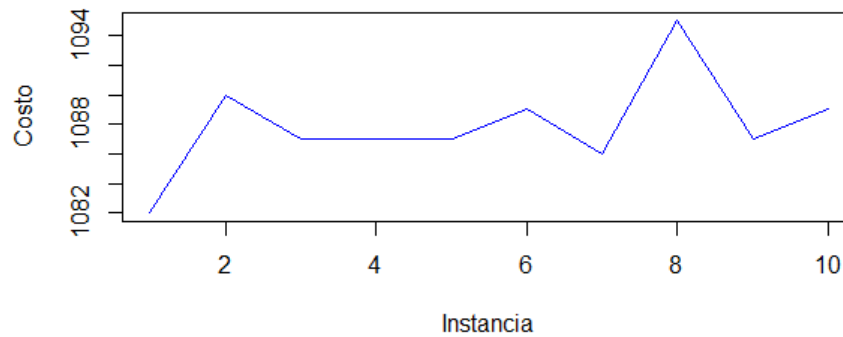


Figura 8.1: Costo modelo 1 tamaño 20

Se obtuvo una eficiencia promedio del 91 %, calculada entre el total de zonas trabajadas frente al objetivo, el cual equivalía al 30 % del total de puntos en la cuadrícula.

La gráfica 8.2 muestra la eficiencia promedio por réplica para el tamaño 20:

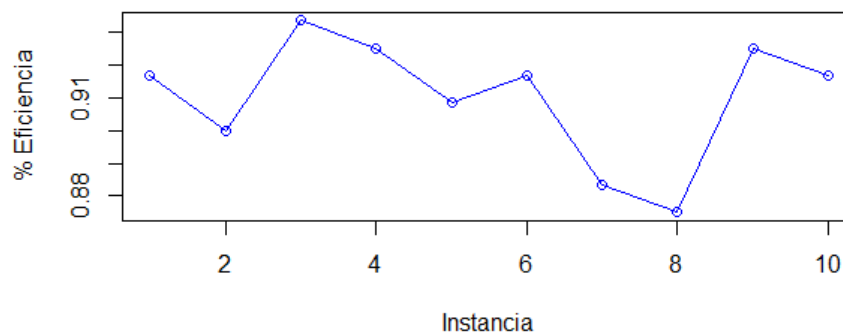


Figura 8.2: Eficiencia modelo 1 tamaño 20

■ Instancia igual a 30

La gráfica 8.3 muestra el costo obtenido al trabajar en el terreno, con un valor mínimo de 1056 y un máximo de 1062:

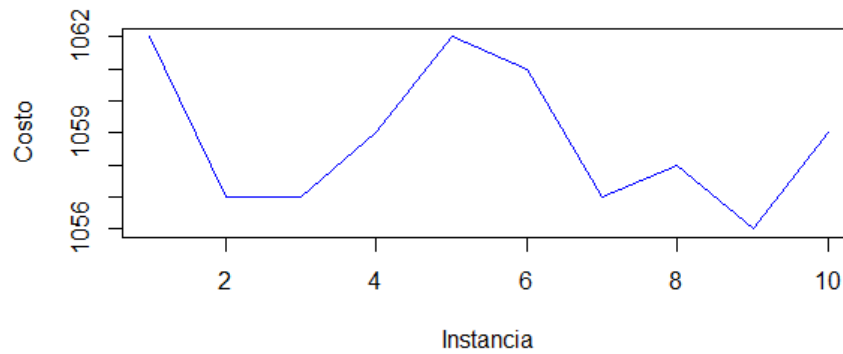


Figura 8.3: Costo modelo 1 tamaño 30

La eficiencia promedio fue del 50.89 %. Para este tamaño, las zonas a optimizar corresponden también al 30 % del total de puntos de la cuadrícula.

La gráfica 8.4 muestra la eficiencia promedio por réplica para el tamaño 30:

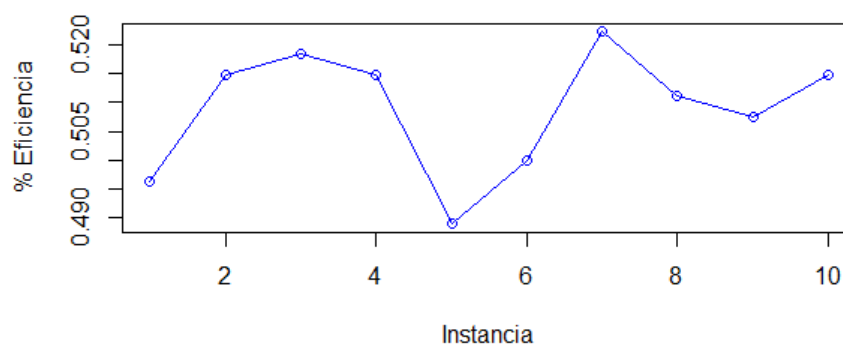


Figura 8.4: Eficiencia modelo 1 tamaño 30

8.2 SEGUNDO MODELO

En este modelo se realizaron 10 réplicas para cada tamaño de instancia (20 y 30).

■ Instancia igual a 20

La gráfica 8.5 muestra el costo obtenido al trabajar en el terreno, con un valor mínimo de 1087 y un máximo de 1092:

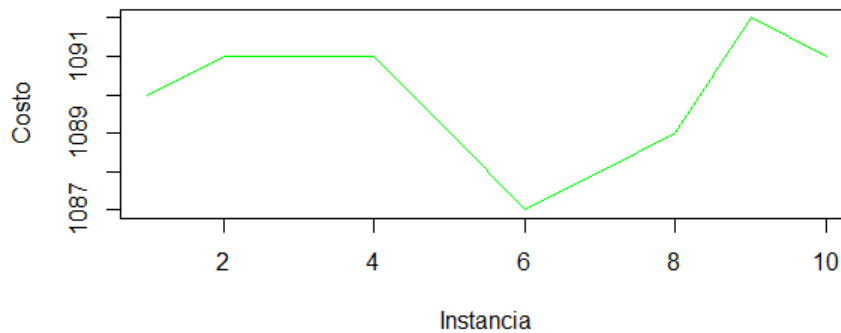


Figura 8.5: Costo modelo 2 tamaño 20

La eficiencia promedio fue de 91.17% sobre las zonas a trabajar, correspondiendo estas al 30 % del total de puntos en la cuadrícula.

La gráfica 8.6 muestra la eficiencia promedio por réplica para el tamaño 20:

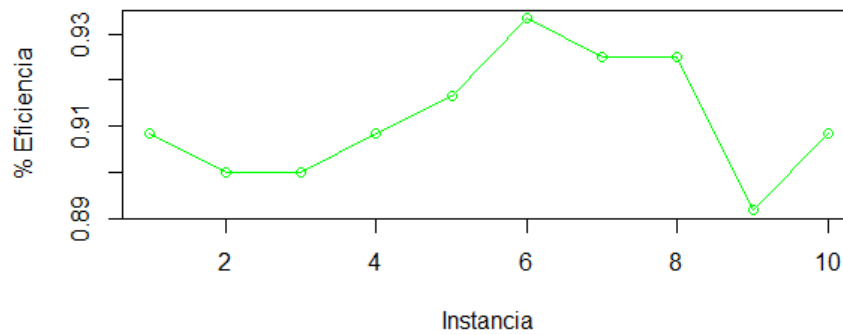


Figura 8.6: Eficiencia modelo 2 tamaño 20

■ Instancia igual a 30

La gráfica 8.7 muestra el costo obtenido al trabajar en el terreno, con un valor mínimo de 1034 y un máximo de 1041:

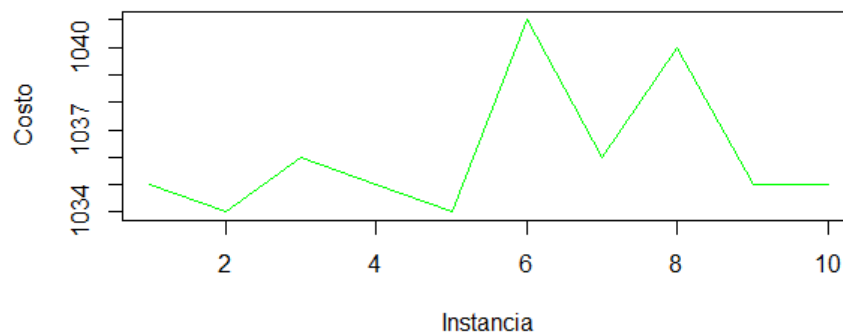


Figura 8.7: Costo modelo 2 tamaño 30

La eficiencia promedio fue del 59.59 %. Para este tamaño, las zonas a optimizar corresponden también al 30 % del total de puntos de la cuadrícula.

La gráfica 8.8 muestra la eficiencia promedio por réplica para el tamaño 30:

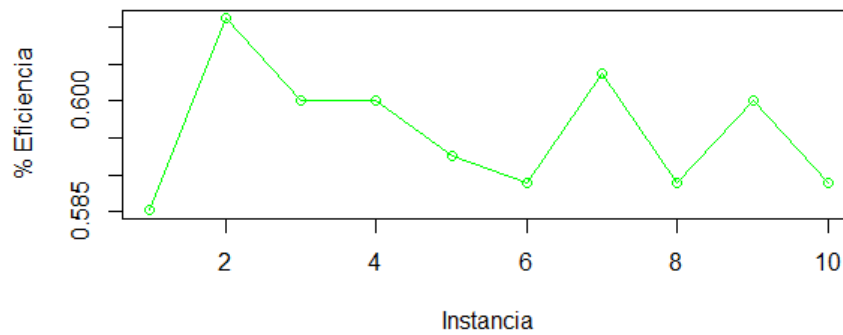


Figura 8.8: Eficiencia modelo 1 tamaño 30

8.3 TERCER MODELO

En este modelo se realizaron 10 réplicas para cada tamaño de instancia (20 y 30).

- **Instancia igual a 20**

La gráfica 8.9 muestra el costo obtenido al trabajar en el terreno, con un valor mínimo de 1075 y un máximo de 1084:

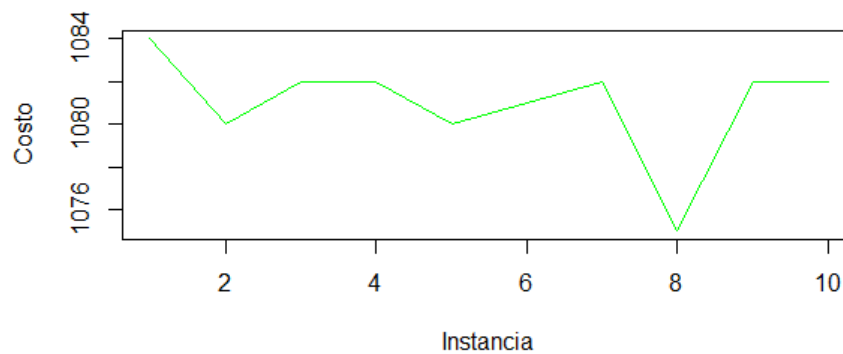


Figura 8.9: Costo modelo 3 tamaño 20

En el tamaño de 20, la eficiencia promedio fue de 98.50 % sobre las zonas a trabajar, correspondiendo estas al 30 % del total de puntos en la cuadrícula.

La gráfica 8.10 muestra la eficiencia promedio por réplica para el tamaño 20:

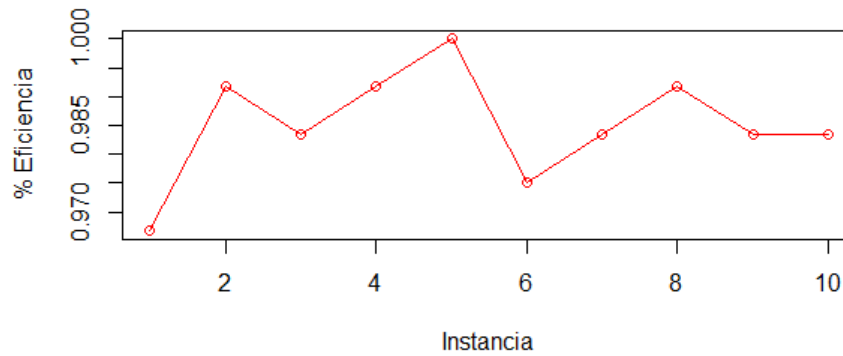


Figura 8.10: Eficiencia modelo 3 tamaño 20

■ Instancia igual a 30

La gráfica 8.11 muestra el costo obtenido al trabajar en el terreno, con un valor mínimo de 1000 y un máximo de 1003:

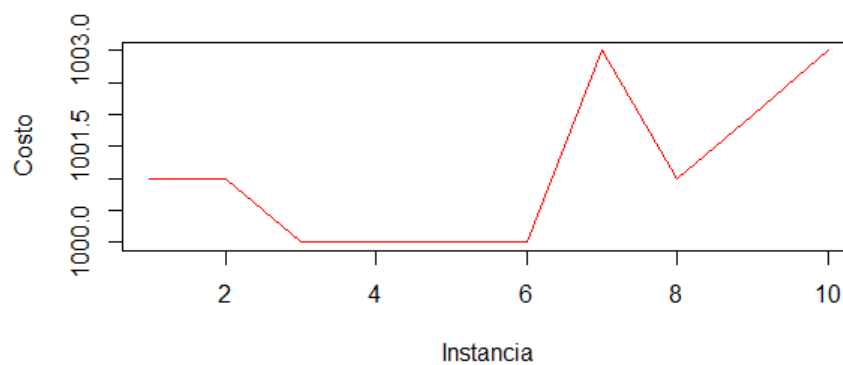


Figura 8.11: Costo modelo 3 tamaño 30

Para el tamaño de 30, la eficiencia promedio fue del 72.52 %. Para este tamaño, las zonas a optimizar corresponden también al 30 % del total de puntos de la

cuadrícula.

La gráfica 8.12 muestra la eficiencia promedio por réplica para el tamaño 30:

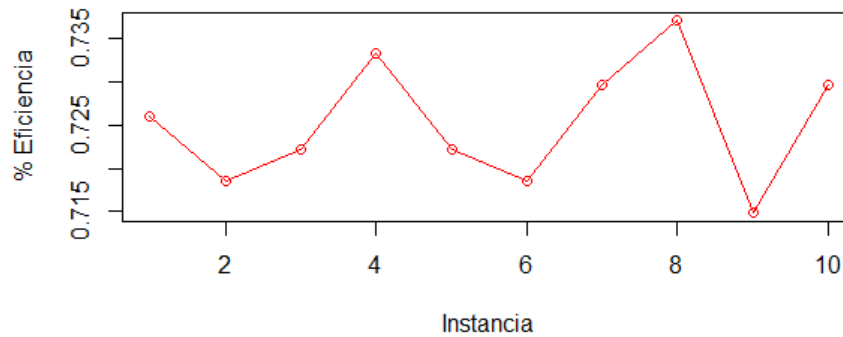


Figura 8.12: Eficiencia modelo 3 tamaño 30

8.4 ANÁLISIS GENERAL

Luego de correr las diferentes réplicas para los tres modelos propuestos y sus correspondiente tamaño de instancia, es necesario comprobar a través de la estadística cual solución es la mejor para el problema propuesto. Para esto, se desarrolló una regresión lineal general y a través del análisis tanto del costo total incurrido como del trabajo total realizado, se comparó cada variable contra el modelo y el tamaño.

■ Costo vs modelo y tamaño

Al realizar una regresión lineal y su correspondiente ANOVA, se evidencia que existe una alta correlación entre el costo, el modelo trabajado y el tamaño de la cuadrícula usado (como se aprecia en la tabla 8.1), dado el valor F que se obtiene para cada variable

Ahora bien, es necesario comprobar si los datos presentan un comportamiento normal para dar por válido la prueba ANOVA. Esto se hizo mediante un

	GL	Sum $error^2$	Med $error^2$	valor F	Pr(>F)
Modelo	2	10881.43	5440.72	1037.42	0.0000
Tamaño	1	44173.07	44173.07	8422.83	0.0000
Modelo:Tamaño	2	6453.23	3226.62	615.24	0.0000
Residuales	54	283.20	5.24		

Tabla 8.1: ANOVA costo vs modelo y tamaño

histograma de frecuencia (8.13) y la prueba Shapiro, donde se obtuvo un valor $p = 0.009030$, el cual es menor a 0.05 y por lo cual se comprueba que los datos correspondiente al costo no siguen una distribución normal. Dado este hallazgo y con el fin de corroborar la correlación entre las variables, se realizó una prueba no paramétrica: prueba Kruskal Wallis, donde se obtuvo un valor $p = 0.007964$ para la relación entre el costo y el modelo, y de $p = 2.68e-11$ para la relación entre el costo y el tamaño. Con estos valores se comprueba que existe relación entre las variables y se da por aceptado el modelo de regresión propuesto.

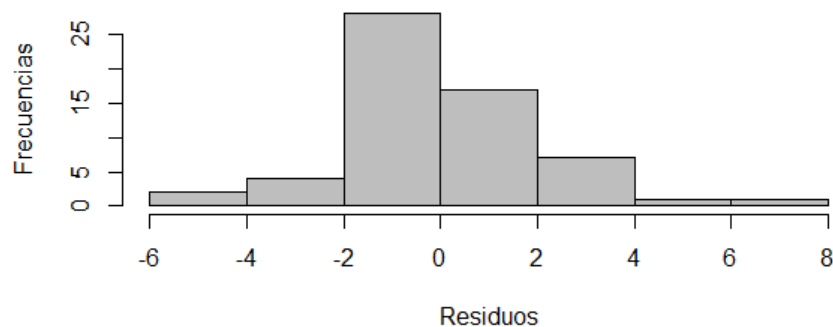


Figura 8.13: Histograma de los residuos de: costo vs modelo y tamaño

El gráfico 8.14 muestra la alta interacción que existe entre el costo, el modelo y el tamaño dado el cruce que hacen las rectas, aunado al valor p de la interacción de la tabla 8.1:

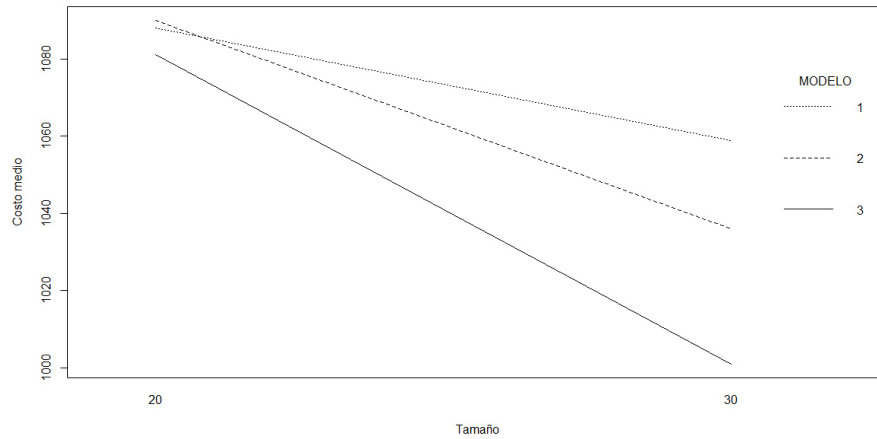


Figura 8.14: interacción costo vs modelo vs tamaño

■ Trabajo vs modelo y tamaño

El trabajo hace referencia al total de zonas tratadas por los drones en cada réplica. Se comprobó mediante una regresión lineal y una prueba de varianza, si existe una relación entre el trabajo, el modelo y el tamaño de instancia. Al efectuar el ejercicio, se comprueba que existe una alta significancia entre estas variables (verificado a través del valor F obtenido para cada variable) como se puede apreciar en la tabla 8.2 :

	GL	Sum <i>error</i> ²	Med <i>error</i> ²	valor F	Pr(>F)
Modelo	2	11690.23	5845.12	1357.58	0.0000
Tamaño	1	41238.82	41238.82	9578.05	0.0000
Modelo:Tamaño	2	6107.43	3053.72	709.25	0.0000
Residuales	54	232.50	4.31		

Tabla 8.2: ANOVA Trabajo vs modelo y tamaño

Continuando con las comprobaciones para corroborar el ANOVA, se realizó la prueba Shapiro y un histograma de frecuencias (figura 8.15) para verificar la normalidad de la variable trabajo. La prueba Shapiro arroja un valor $p = 0.6712$, el cual es mayor a 0.05 y por lo cual se infiere que la variable sigue una distribución normal y se comprueba que el ANOVA fue correcto para el

modelo planteado y existe por lo tanto, alta significancia entre las variables propuestas.

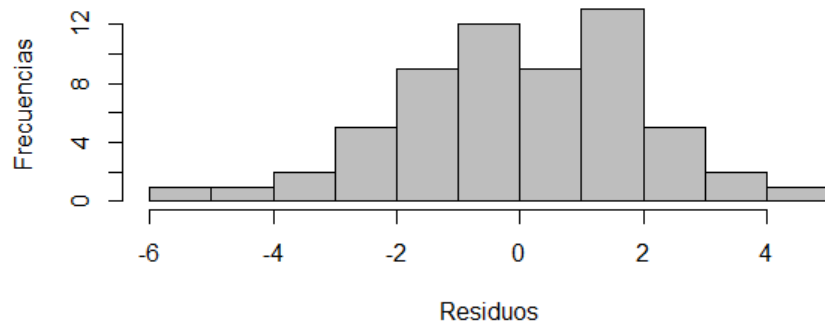


Figura 8.15: Histograma residuos trabajo vs modelo y tamaño

El gráfico 8.16 muestra la alta interacción que existe entre el trabajo, el modelo y el tamaño dado el cruce que hacen las rectas:

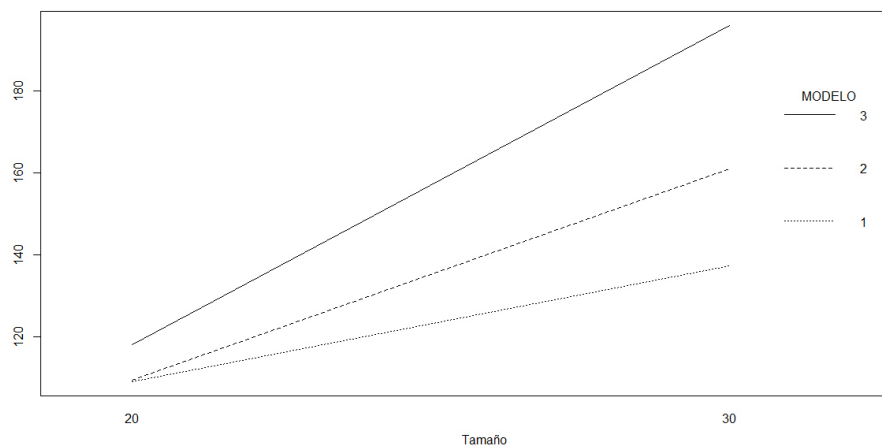


Figura 8.16: interacción trabajo vs modelo vs tamaño

A partir de los gráficos de interacción para las variables costo y trabajo, se evidencia que presentan comportamientos inversamente proporcionales, es decir, la disminución en el costo producirá un aumento en el trabajo y viceversa, siendo el modelo 3 el que presenta la mejor relación trabajo-costo. No obstante, para apoyar

esta aseveración, se crearon dos gráficos de superficie, donde se proyectan las tres variables estudiadas en cada caso y así poder dar una evaluación racional del mejor modelo para solucionar el problema:

- Para la relación costo vs modelo y tamaño se construyó el gráfico 8.17, donde es claro que la disminución del costo conlleva a emplear un modelo más alto (ordenados 1-2-3) y a un mayor tamaño de la instancia.

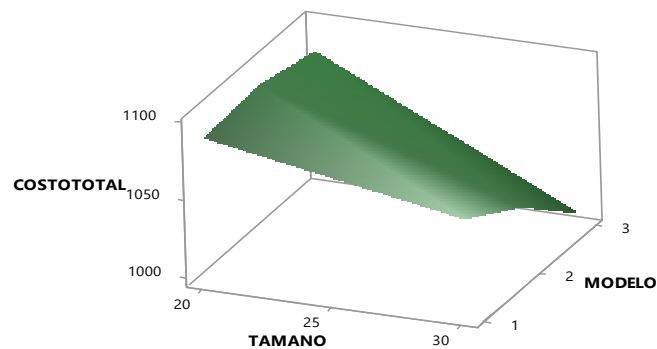


Figura 8.17: Costo vs tamaño y modelo

- Para la relación trabajo vs modelo y tamaño se construyó el gráfico 8.18, donde es claro que el aumento de las zonas trabajadas conlleva a emplear un modelo más alto (ordenados 1-2-3) y a un mayor tamaño de la instancia.

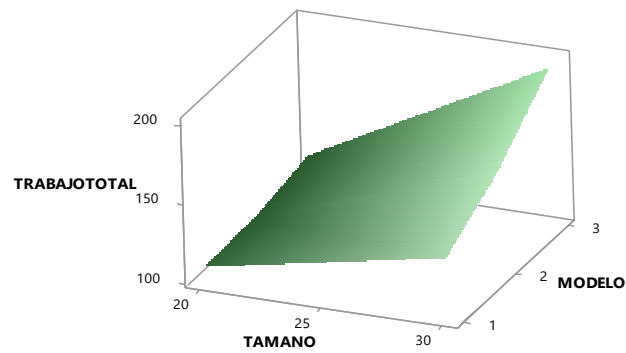


Figura 8.18: Trabajo vs tamaño y modelo

Dados los resultados anteriores, se puede observar que el modelo 3 presenta las condiciones necesarias para considerarse como una buena solución al problema planteado, ya que para el costo representa una disminución y para el trabajo un aumento de las zonas tratadas.

CAPÍTULO 9

CONCLUSIONES

*La vida es el arte de sacar conclusiones suficientes
a partir de datos insuficientes.*

Samuel Butler

9.1. Conclusiones	56
9.2. Trabajo futuro	57

9.1 CONCLUSIONES

En esta investigación se puede concluir que al comparar los tres modelos planteados para solucionar el problema de optimización, se tiene suficiente evidencia estadística para afirmar que el modelo número 3 presenta las condiciones exigidas para brindar una solución acorde a las restricciones y necesidades planteadas.

El uso de búsqueda tabú en el modelo 3, permitió que este fuera el que permitiera las mejores soluciones, además de la comunicación que se brindó entre los drones para informar las posiciones visitadas y alimentar la lista tabú.

La búsqueda local y el algoritmo de ascenso de colinas presentaron comporta-

mientos similares, a pesar de que el último tenía un componente *greedy* para seleccionar las mejores posiciones; aún así se estancaba en óptimos locales. Para poder corregir este inconveniente, sería interesante implementar ideas de recocido simulado para este caso.

Se pudo comprobar la significancia que existe entre el modelo usado, el tamaño de instancia, el costo y el trabajo realizado; siendo estos dos últimos inversamente proporcionales, lo que nos da un problema mono-objetivo, donde un aumento o disminución en el la variable trabajo o costo, repercutirá inmediatamente en la otra.

La comunicación entre los agentes juega un papel clave en la solución del problema, puesto que permite la transmisión de aquellas posiciones visitadas y su exclusión para evitar la redundancia de visita de zonas por otro agente.

Es interesante el entorno social generado por los drones y la interacción que se presenta para alcanzar los objetivos comunes, mediante planes concisos, metas claramente definidas y el uso de metaheurísticas para mejorar el desplazamiento.

9.2 TRABAJO FUTURO

Desarrollar estrategias basadas en los problemas del agente viajero (*travelling Salesman Problem*, TSP) y enrutamiento de vehículos (*Vehicle Routing Problem*, VRP) para comparar el desempeño de estas estrategias con la propuesta en esta tesis.

Incluir aprendizaje a los agentes, para que estos mejoren a partir de la experiencia el trabajo y los movimientos realizados.

Implementar nuevos planes que permitan al dron recordar la última posición visitada, para que al recargar pueda volver a ella y continuar el trabajo desde ese punto.

Probar un nuevo modelo donde se implemente un lider, el cual guie el trabajo en el terreno, divida las zonas y controle los puntos donde los agentes deben moverse.

APÉNDICE A

TABLAS DISEÑO SISTEMA MULTIAGENTE

EN este apéndice se muestran las tablas que contienen la estructura del sistema multiagente diseñado bajo la metodología Prometheus. Para su mejor comprensión se ha dividido según las fases: especificación del sistema, diseño arquitectónico y diseño detallado.

A.1 FASE ESPECIFICACIÓN DEL SISTEMA

A.1.1 OBJETIVOS

- La tabla A.1 muestra el objetivo *recargar batería* con sus correspondientes subobjetivos:

Objetivo		
Nombre	Recargar batería	
Descripción	Mantener batería para efectuar las tareas	
Pasos		
Nombre	Descripción	Datos
Conocer nivel batería	Determinar el nivel de carga	Cantidad recargas
Mantener batería carga- da	Permanecer con batería necesaria	

Tabla A.1: Recargar batería

- La tabla A.2 muestra el objetivo *controlar la humedad* con sus correspondientes subobjetivos:

Objetivo		
Nombre	Controlar la humedad	
Descripción	Suministrar la cantidad de agua necesaria.	
Pasos		
Nombre	Descripción	Datos
Identificar zonas poca humedad	Según las coordenadas recibidas, definir las zonas a irrigar.	Coordenadas
Irigar zonas poca humedad	Suministrar agua a las zonas que lo requieran.	

Tabla A.2: Controlar la humedad

- La tabla A.3 muestra el objetivo *abonar el terreno* con sus correspondientes subobjetivos:

Objetivo		
Nombre	Abonar el terreno	
Descripción	Suministrar fertilizantes y agroquímicos.	
Pasos		
Nombre	Descripción	Datos
Identificar zonas faltas nutrientes	Según las coordenadas recibidas, definir las zonas a abonar.	Coordenadas
Fertilizar zonas faltas nutrientes	Suministrar abono a las zonas que lo requieran.	

Tabla A.3: Controlar la humedad

- La tabla A.4 muestra el objetivo *controlar plagas* con sus correspondientes subobjetivos:

Objetivo		
Nombre	Controlar plagas	
Descripción	Detectar de forma oportuna las plagas.	
Pasos		
Nombre	Descripción	Datos
Identificar zonas con plagas	Según las coordenadas recibidas, definir las zonas a fumigar.	Coordenadas
Fumigar zonas con plagas	Suministrar pesticidas a las zonas que lo requieran.	

Tabla A.4: Controlar las plagas

A.1.2 ROLES

- La tabla A.5 muestra el rol *rolOptimizar*:

Rol	
Nombre	<i>rolOptimizar</i>
Descripción	Es responsable de controlar el nivel de humedad, abonar el terreno y el control de plagas.
Acciones	<i>mover(zona), optimizar(zona), moverAleatorio</i>
Objetivos	<i>Abonar el terreno, controlar la humedad, controlar plagas</i>

Tabla A.5: Rol Optimizar

- La tabla A.6 muestra el rol *rolBateria*:

Rol	
Nombre	<i>rolBateria</i>
Descripción	Gestiona el proceso de recarga de la batería para los drones.
Acciones	<i>moverHacia(base), recargar(batería)</i>
Objetivos	<i>Recargar batería</i>

Tabla A.6: Rol Bateria

A.1.3 ESCENARIOS

- La tabla A.7 muestra el escenario *Recargar escenario*:

Escenario			
Nombre	Recargar Escenario		
Descripción	Recargar la batería a los drones		
Evento	Batería baja		
Pasos			
Tipo	Nombre	Rol	Datos
Percepción	<i>batería(nivel)</i>	rolRecargar	cantidad recargas
Percepción	<i>agente(x,y)</i>	rolRecargar	
Percepción	<i>base(x,y)</i>	rolRecargar	
Percepción	<i>pasos</i>	rolRecargar	
Objetivo	Recargar batería	rolRecargar	
Acción	<i>moverHacia(base)</i>	rolRecargar	Tamaño del terreno
Acción	<i>recargar(batería)</i>	rolRecargar	

Tabla A.7: Recharge scenario

- La tabla A.8 muestra el escenario *Optimizar escenario*:

Escenario			
Nombre	Optimizar Escenario		
Descripción	Suplir las necesidades del terreno		
Evento	Según agenda		
Pasos			
Tipo	Nombre	Rol	Datos
Percepción	requiereTratamiento(zona)	rolOptimizar	
Percepción	pos(x,y)	rolOptimizar	
Percepción	agente(x,y)	rolOptimizar	
Percepción	gridsize(x,y)	rolOptimizar	
Objetivo	Controlar la humedad	rolOptimizar	
Objetivo	Abonar el terreno	rolOptimizar	
Objetivo	Controlar plagas	rolOptimizar	
Acción	mover(zona)	rolOptimizar	
Acción	optimizar(zona)	rolOptimizar	
Acción	moverAleatorio	rolOptimizar	

Tabla A.8: Optimizar Escenario

A.1.4 PERCEPCIONES

- La tabla A.9 muestra la percepción que indica el nivel de batería:

Percepción	
Nombre	<i>batería(nivel)</i>
Descripción	Evento que indica el nivel de batería
Información	Nivel de batería actual
Conocimiento Actualizado	Batería disponible
Fuente	Interna
Procesamiento	Ninguno
Agentes que responden	<i>drone1, drone2, drone3</i>
Frecuencia	Al ejecutar un movimiento

Tabla A.9: Percepción nivel de batería

- La tabla A.10 muestra la percepción que indica si la posición siguiente está ocupada por otro dron:

Percepción	
Nombre	<i>agente(x,y)</i>
Descripción	Indica si la posición está ocupada
Información	posición (x,y) libre/ocupada
Conocimiento Actualizado	Posición disponible/no disponible
Fuente	Interna
Procesamiento	Ninguno
Agentes que responden	<i>drone1, drone2, drone3</i>
Frecuencia	cada movimiento

Tabla A.10: Percepción otro dron

- La tabla A.11 muestra la percepción que indica la posición de la base para recargar:

Percepción	
Nombre	$base(x,y)$
Descripción	Indica donde se encuentran las bases
Información	Base en posición (x,y)
Conocimiento Actualizado	Base para recargar
Fuente	Interna
Procesamiento	Ninguno
Agentes que responden	$drone1, drone2, drone3$
Frecuencia	Al inicio del proceso

Tabla A.11: Percepción base para recargar

- La tabla A.12 muestra la percepción que indica los movimientos disponibles para avanzar por el terreno (1 movimiento es igual a 1 % de batería), estos pueden ser menores al total de la carga:

Percepción	
Nombre	$pasos$
Descripción	Indica los movimientos disponibles
Información	Movimientos restantes
Conocimiento Actualizado	Total de movimientos
Fuente	Interna
Procesamiento	Ninguno
Agentes que responden	$drone1, drone2, drone3$
Frecuencia	Al inicio del proceso

Tabla A.12: Percepción movimientos disponibles

- La tabla A.13 muestra la percepción que indica si la posición requiere algún tratamiento:

Percepción	
Nombre	<i>requiereTratamiento(zona)</i>
Descripción	Indica si la posición requiere atención
Información	Sitio que requiere tratamiento
Conocimiento Actualizado	Coordenadas para trabajar
Fuente	Interna
Procesamiento	Ninguno
Agentes que responden	<i>drone1, drone2, drone3</i>
Frecuencia	Antes de cada movimiento

Tabla A.13: Percepción zona para tratamiento

- La tabla A.14 muestra la percepción que indica la posición actual:

Percepción	
Nombre	<i>pos(x,y)</i>
Descripción	Indica la posición actual
Información	Posición actual del dron
Conocimiento Actualizado	Posición disponible
Fuente	Tamaño del terreno
Procesamiento	Ninguno
Agentes que responden	<i>drone1, drone2, drone3</i>
Frecuencia	Después de cada movimiento

Tabla A.14: Percepción posición actual

- La tabla A.15 muestra la percepción que indica el tamaño de la cuadrícula:

Percepción	
Nombre	<i>tamañoCuadrícula(x,y)</i>
Descripción	Indica el tamaño del terreno
Información	Dimensiones del terreno
Conocimiento Actualizado	Dimensiones
Fuente	Tamaño del terreno
Procesamiento	Ninguno
Agentes que responden	<i>drone1, drone2, drone3</i>
Frecuencia	Al inicio del proceso

Tabla A.15: Percepción tamaño del terreno

A.1.5 ACCIONES

- La tabla A.16 muestra la acción para mover a posición que requiere atención:

Acción	
Nombre	<i>mover(zona)</i>
Descripción	Mover a zona requerida
Parámetros	Verificar si posición siguiente está libre
Duración	Proceso de optimización
Fallo	Información no disponible
Efectos Laterales	Ninguno

Tabla A.16: Acción avanzar a zona para trabajar

- La tabla A.17 muestra la acción para optimizar el terreno:

Acción	
Nombre	<i>optimizar(zona)</i>
Descripción	Optimiza la posición actual
Parámetros	Verificar si la zona requiere tratamiento
Duración	Hasta finalizar tarea
Fallo	Movimiento no disponible
Efectos Laterales	Ninguno

Tabla A.17: Acción optimizar área

- La tabla A.18 muestra la acción para moverse por el terreno:

Acción	
Nombre	<i>moverAleatorio</i>
Descripción	Mueve hacia posición libre
Parámetros	Verificar si posición siguiente está libre
Duración	Hasta hallar posición libre
Fallo	Movimiento no disponible
Efectos Laterales	Ninguno

Tabla A.18: Acción moverse por el terreno

La tabla A.19 muestra la acción para moverse hacia la base:

Acción	
Nombre	<i>moverHacia(base)</i>
Descripción	Desplazarse hacia la base
Parámetros	Ninguno
Duración	Según la distancia a la base
Fallo	Obstáculo en el camino
Efectos Laterales	Ninguno

Tabla A.19: Acción mover a base

- La tabla A.20 muestra la acción para recargar la batería:

Acción	
Nombre	<i>recargar(batería)</i>
Descripción	Recargar la Batería
Parámetros	Verificar si se encuentra en la base
Duración	Hasta recarga completa
Fallo	Base ocupada
Efectos Laterales	Ninguno

Tabla A.20: Acción recargar la batería

A.2 FASE DISEÑO ARQUITECTÓNICO

A.2.1 IDENTIFICACIÓN DE LOS AGENTES

- La tabla A.21 muestra el agente *drone1*, el cual es el encargado de proporcionar agua según requerimientos:

Agente			
Nombre	<i>drone1</i>		
Descripción	Suministra agua a sitios específicos		
Cardinalidad mínima	1	Cardinalidad máxima	inf
Duración	Ilimitada	Inicialización	Al cargar el mapa
En caso de fallo	Nada	Percepciones	<i>pos(x,y), needsIrrigate(zone), gridzise(x,y), agente(x,y), batería(nivel), base(x,y), pasos</i>
Acciones	<i>mover(zona), optimizar(zona), moverAleatorio, moverHacia(base), recargar(batería), detener</i>	Usa datos	tamaño terreno
Produce datos	lista restringida posiciones, posiciones irrigadas	Datos internos	Por definir
Objetivos	<i>Controlar la Humedad</i>	roles	<i>rolOptimizar</i>

Tabla A.21: Agente drone1

- La tabla A.22 muestra el agente *drone2*, el cual es el encargado de fertilizar el terreno:

Agente			
Nombre	<i>drone2</i>		
Descripción	Suministra fertilizante a sitios específicos		
Cardinalidad mínima	1	Cardinalidad máxima	inf
Duración	Ilimitada	Inicialización	Al cargar el mapa
En caso de fallo	Nada	Percepciones	<i>pos(x,y), needsFertilize(zone), gridzise(x,y), agente(x,y), batería(nivel), base(x,y), pasos</i>
Acciones	<i>mover(zona), optimizar(zona), moverAleatorio, moverHacia(base), recargar(batería), detener</i>	Usa datos	tamaño terreno
Produce datos	lista restringida posiciones, posiciones fertilizadas	Datos internos	Por definir
Objetivos	<i>Abonar el terreno</i>	roles	<i>rolOptimizar</i>

Tabla A.22: Agente drone2

- La tabla A.23 muestra el agente *drone3*, el cual es el encargado de controlar las plagas:

Agente			
Nombre	<i>drone3</i>		
Descripción	Controla las plagas		
Cardinalidad mínima	1	Cardinalidad máxima	inf
Duración	Ilimitada	Inicialización	Al cargar el mapa
En caso de fallo	Nada	Percepciones	<i>pos(x,y), needsFertilize(zone), gridzise(x,y), agente(x,y), batería(nivel), base(x,y), pasos</i>
Acciones	<i>mover(zona), optimizar(zona), moverAleatorio, moverHacia(base), recargar(batería), detener</i>	Usa datos	tamaño terreno
Produce datos	lista restringida posiciones, posiciones fumigadas	Datos internos	Por definir
Objetivos	<i>Controlar las plagas</i>	roles	<i>rolOptimizar</i>

Tabla A.23: Agente drone3

A.2.2 DATOS

A.2.3 PROTOCOLOS DE COMUNICACIÓN

- El protocolo *Irrigar* indica que se ha suministrado agua en un sitio específico por parte del *drone1*; este envía un mensaje con las coordenadas del sitio a los demás agentes, los cuales deciden si las incluyen en la lista de posiciones visitadas; para de esta forma, evitar un doble paso por una misma área . Este proceso se describe en la tabla A.24:

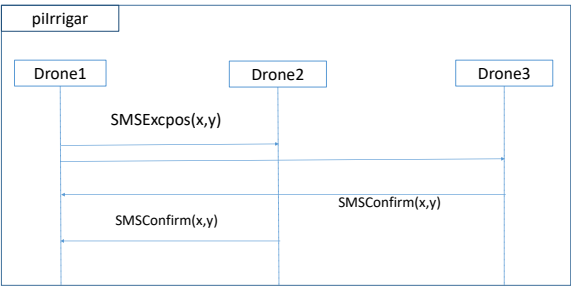
Protocolo	
Nombre	<i>Irrigar</i>
Descripción	Enviar coordenadas con sitio visitado para su exclusión
Mensajes	<i>SMSExcpos(x,y)</i> , <i>SMSConfirmar(x,y)</i>
Escenarios	<i>Optimizar Escenario</i>
Agentes	<i>drone1, drone2, drone3</i>
Representación	
Notas	

Tabla A.24: Protocolo irrigar

- El protocolo *Abonar* especifica que se ha trabajado en un sitio específico y por ende ya no debe ser visitado; para ello, el *drone2* envía un mensaje con las coordenadas del sitio al *drone1* y *drone3*, las cuales deben ser aceptadas o rechazadas por los agentes. Este proceso se describe en la tabla A.25:

Protocolo	
Nombre	<i>Abonar</i>
Descripción	Enviar coordenadas con sitio visitado para su exclusión
Mensajes	<i>SMSExcpos(x,y)</i> , <i>SMSConfirmar(x,y)</i>
Escenarios	<i>Optimizar Escenario</i>
Agentes	<i>drone1</i> , <i>drone2</i> , <i>drone3</i>
Representación	<pre>sequenceDiagram participant Drone1 participant Drone2 participant Drone3 Drone2->>Drone1: SMSExcpos(x,y) Drone2->>Drone3: SMSExcpos(x,y) Drone1->>Drone2: SMSConfirm(x,y) Drone3->>Drone2: SMSConfirm(x,y)</pre>
Notas	

Tabla A.25: Protocolo abonar

- El protocolo *Fumigar* especifica las coordenadas donde el *drone3* ha suministrado plagicidas; este envía un mensaje con las coordenadas del sitio al *drone1* y *drone2*, los cuales deciden si agregarlas o no en la lista restringida de sitios a visitar. Este proceso se describe en la tabla A.26:

Protocolo	
Nombre	<i>Fumigar</i>
Descripción	Enviar coordenadas con sitio visitado para su exclusión
Mensajes	$SMSExcpos(x,y)$, $SMSConfirmar(x,y)$
Escenarios	<i>Optimizar Escenario</i>
Agentes	<i>drone1, drone2, drone3</i>
Representación	<pre> sequenceDiagram participant Drone1 participant Drone2 participant Drone3 Drone1->>Drone2: SMSExcpos(x,y) Drone1->>Drone3: SMSExcpos(x,y) Drone2->>Drone1: SMSConfirm(x,y) Drone3->>Drone1: SMSConfirm(x,y) </pre>
Notas	

Tabla A.26: Protocolo fumigar

A.2.4 MENSAJES

- La tabla A.27 contiene el mensaje que el *drone1* le envía a los agentes *drone2* y *drone3*:

Mensaje	
Nombre	$SMSExcpos(x,y)$
Descripción	Enviar coordenadas de sitio trabajado
distribución	De <i>drone1</i> a <i>drone2</i> y <i>drone3</i>
Propósito	Indicar zonas visitadas
Información contenida	Coordenadas de punto visitado

Tabla A.27: Mensaje irrigar

- La tabla A.28 contiene el mensaje que el *drone2* y el *drone3* le envía al *drone1*:

Mensaje	
Nombre	<i>SMSConfirmar(x,y)</i>
Descripción	Confirma recepción de coordenadas
distribución	De <i>drone2</i> y <i>drone3</i> a <i>drone1</i>
Propósito	Confirmación de recepción
Información contenida	Aceptación o negación

Tabla A.28: Respuesta irrigar

- La tabla A.29 contiene el mensaje que el *drone2* le envía al *drone1* y *drone3*:

Mensaje	
Nombre	<i>SMSExcpos(x,y)</i>
Descripción	Enviar coordenadas de sitio trabajado
distribución	De <i>drone2</i> a <i>drone1</i> y <i>drone3</i>
Propósito	Indicar zonas visitadas
Información contenida	Coordenadas de punto visitado

Tabla A.29: Mensaje abonar

- La tabla A.30 contiene el mensaje que el *drone1* le envía al *drone2* y *drone3*:

Mensaje	
Nombre	<i>SMSConfirmar(x,y)</i>
Descripción	Confirma recepción de coordenadas
distribución	De <i>drone1</i> y <i>drone3</i> a <i>drone2</i>
Propósito	Confirmación de recepción
Información contenida	Aceptación o negación

Tabla A.30: Respuesta abonar

- La tabla A.31 contiene el mensaje que el *drone3* le envía al *drone1* y *drone2*:

Mensaje	
Nombre	$SMSExcpos(x,y)$
Descripción	Enviar coordenadas de sitio trabajado
distribución	De <i>drone3</i> a <i>drone1</i> y <i>drone2</i>
Propósito	Indicar zonas visitadas
Información contenida	Coordenadas de punto visitado

Tabla A.31: Mensaje fumigar

- La tabla A.32 contiene el mensaje que el *drone1* y *drone2* le envían al *drone3*:

Mensaje	
Nombre	$SMSConfirmar(x,y)$
Descripción	Confirma recepción de coordenadas
distribución	De <i>drone1</i> y <i>drone2</i> a <i>drone3</i>
Propósito	Confirmación de recepción
Información contenida	Aceptación o negación

Tabla A.32: Respuesta fumigar

A.3 FASE DISEÑO DETALLADO

A.3.1 CAPACIDADES

La primera capacidad (tabla A.33) hace referencia al evento de recargar la batería, definiendo de igual manera el plan asociado:

Capacidad	
Nombre	<i>Recargar</i>
Descripción	Mantener el nivel de batería suficiente para trabajar
Objetivos	Recargar batería
Protocolos	Ninguno
Mensajes Entrantes	Ninguno
Mensajes Salientes	Ninguno
Mensajes Internos	<i>Mi batería está baja, sin batería</i>
Percepciones	<i>agente(x,y), base(x,y), batería(nivel), pasos</i>
Acciones	<i>moverHacia(base), recargar(batería), detener</i>
Datos usados	<i>Ubicación bases</i>
Datos producidos	<i>Cantidad de recargas</i>
Datos internos	nivel de batería
Planes incluidos	<i>revisar(zona), irBase, parar, apagar</i>
Capacidades incluidas	Ninguna

Tabla A.33: Capacidad recargar

La segunda capacidad (tabla A.34) se refiere al evento de optimizar el terreno:

Capacidad	
Nombre	<i>Optimizar</i>
Descripción	Trabajar en los sitios que lo requieran
Objetivos	Controlar la humedad, abonar el terreno, controlar plagas
Protocolos	<i>Irigar, Abonar, Fumigar</i>
Mensajes Entrantes	<i>SMSConfirmar(x,y)</i>
Mensajes Salientes	<i>SMSExcpos(x,y)</i>
Mensajes Internos	Ninguno
Percepciones	<i>requiereTratamiento(zona), agente(x,y), gridzise(x,y), pos(x,y), batería(nivel), pasos</i>
Acciones	<i>mover(zona), optimizar(zona), moverAleatorio, detener</i>
Datos usados	Tamaño del terreno
Datos producidos	Lista restringida posiciones, posiciones optimizadas
Datos internos	nivel de batería, movimientos disponibles
Planes incluidos	<i>trabajar, parar</i>
Capacidades incluidas	Ninguna

Tabla A.34: Capacidad optimizar

A.3.2 PLANES

El plan descrito para revisar el nivel de batería (tabla A.35), describe el procedimiento en el cual, se define la ruta a seguir, haciendo la posición inicial del agente la estación de carga y cada vez que se avanza, el agente revisa si cuenta con el nivel de batería adecuado. En caso afirmativo, se activa el plan para trabajar en la zona actual y luego continua con la revisión; en caso contrario, el agente se dirige a la base a recargar la batería y se repite el proceso.

Plan	
Nombre	<i>revisar(zona)</i>
Descripción	Verificar el nivel de batería y movimientos disponibles para moverse y realizar el trabajo
Iniciador	<i>batería(nivel), pasos</i>
Mensajes Entrantes	Ninguno
Mensajes Salientes	Ninguno
Datos usados	tamaño del terreno
Datos producidos	Ninguno
Fallo	Falta de acceso a los datos
Recuperación de fallo	Envíar mensaje a usuario
Procedimiento	<pre> INICIO batería(nivel) = 100; pasos = n; Mientras se tenga batería disponible Si batería(nivel) > dist(base) Activar plan trabajar; De lo contrario si batería(nivel) <= dist(base) Activar plan irBase; De lo contrario Activar plan parar; Fin si Fin Mientras FIN </pre>

Tabla A.35: Plan revisar(zona)

El plan descrito para ir a la base (tabla A.36), describe el procedimiento en el cual, el drone se dirige a recargar al momento de tener un nivel bajo de su batería. Se activa en el plan *revisar(zona)* al medir la distancia entre su posición actual y la base más cercana (la cual debe ser menor o igual al valor restante de la batería, para llegar antes de tener nivel 0 de carga).

Plan	
Nombre	<i>irBase</i>
Descripción	Dirigirse a la base a recargar la batería
Iniciador	<i>revisar(zona)</i>
Mensajes Entrantes	Ninguno
Mensajes Salientes	Ninguno
Datos usados	tamaño del terreno
Datos producidos	Cantidad de recargas
Fallo	Distancia a base mayor a disponible de carga
Recuperación de fallo	Envíar mensaje a usuario
Procedimiento	<pre> INICIO Obtener posición base (x,y) ; Dist pos(x,y) a base (x,y) = n; Mientras se tenga batería disponible Si pos(x,y) > base (x,y) Imprime "moviendo a destino"; moverHacia; De lo contrario si pos(x,y) = base (x,y) Recargar(batería); De lo contrario si batería(nivel) = 0 Activar plan parar; Fin si Fin Mientras FIN </pre>

Tabla A.36: Plan irBase

El plan para trabajar en el terreno (tabla A.37), incluye la revisión de la posición actual del agente y de las cuatro posiciones adyacentes (norte, sur, este y oeste). Si cualquiera de esas posiciones requiere ser optimizada, el agente se mueve y realiza la tarea correspondiente; en caso de que ninguna posición requiera tratamiento, se moverá a otra posición libre y seguirá verificando hasta terminar su labor.

Plan	
Nombre	<i>trabajar</i>
Descripción	Realizar el trabajo en la zonas requeridas
Iniciador	<i>revisar(zona)</i>
Mensajes Entrantes	<i>SMSConfirmar(x,y)</i>
Mensajes Salientes	<i>SMSExcpos(x,y)</i>
Datos usados	Tamaño del terreno
Datos producidos	Lista restringida posiciones, posiciones optimizadas
Fallo	Falta de acceso a los datos
Recuperación de fallo	Envíar mensaje a usuario
Procedimiento	<pre> INICIO Obtener pos(x,y); Obtener requiereTratamiento(x,y); Mientras haya posiciones sin tratar Si pos(x,y) o requiereTratamiento(x,y) = TRUE Imprime: "mi posición requiere tratamiento"; Optimizar(zona); Enviar SMSExcpos(x,y); Solicitar SMSConfirmar(x,y); De lo contrario si pos(x,y) o requiereTratamiento(x,y) = FALSE Imprime: "la zona no necesita tratamiento"; De lo contrario si bateria(nivel) = 0 Activar plan parar; Fin si Fin Mientras FIN </pre>

Tabla A.37: Plan trabajar

El plan *parar* (tabla A.38) deja al agente en estado de hibernación al tener un nivel de batería igual a cero o al agotar los movimientos disponibles.

Plan	
Nombre	<i>parar</i>
Descripción	Detener al dron
Iniciador	<i>revisar(zona), trabajar, irBase</i>
Mensajes Entrantes	Ninguno
Mensajes Salientes	Ninguno
Datos usados	Ninguno
Datos producidos	Ninguno
Fallo	Falta de acceso a los datos
Recuperación de fallo	Envíar mensaje a usuario
Procedimiento	<pre> INICIO bateria(nivel) = 0; pasos = 0; Si bateria(nivel) = 0 Imprime: "mi bateria está baja"; Parar; De lo contrario si pasos = 0 Imprime: "parando por número máximo de movimientos"; parar; Fin si FIN </pre>

Tabla A.38: Plan parar

BIBLIOGRAFÍA

- [1] Gastón A Addati and Gabriel Pérez Lance. Introducción a los uav's, drones o vants de uso civil. Technical report, Serie Documentos de Trabajo, Universidad del CEMA, 2014.
- [2] Pilar Barreiro Elorza and Constantino Valero Ubierna. Drones en la agricultura. *Tierras de Castilla y León: Agricultura*, 220:36–42, 2014.
- [3] Guerra en Siria, Semana Santa Sevilla, and PSOE Primarias. El uso de drones llega a la agricultura.
- [4] Gerhard Weiss. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 1999.
- [5] Rafael H Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons, 2007.
- [6] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [7] Tomás de Teresa Trancón. Un sistema de control inteligente de entrada/salida de humanos a un recinto siguiendo la metodología vigilagent.
- [8] Lin Padgham and Michael Winikoff. *Developing intelligent agent systems: A practical guide*, volume 13. John Wiley & Sons, 2005.

-
- [9] Lin Padgham, John Thangarajah, and Paresh Paul. Prometheus design tool version 2.5 user manual, 2009.
- [10] Fundación Vasca para la seguridad alimentaria. Berezi@35 drones y su uso en la agricultura. Technical report, Eliko, 2014.
- [11] Emiliano García and Fernando Flego. Agricultura de precisión. *Revista Ciencia y Tecnología. Recuperado de [http://www.palermo.edu/ingenieria/Ciencia_y_Tecnologia/ciencia_y_tecnologia_8.html](http://www.palermo.edu/ingenieria/Ciencia_y_Tecnologia/ciencia_y_tecnologia/ciencia_y_tecnologia_8.html)*, 2008.
- [12] A Barrientos, J Del Cerro, P Gutiérrez, R San Martín, A Martínez, and C Rossi. Vehículos aéreos no tripulados para uso civil. tecnología y aplicaciones. *Universidad politécnica de Madrid, Madrid*, 2007.
- [13] Francisca López-Granados. Uso de vehículos aéreos no tripulados (uav) para la evaluación de la producción agraria. *ambienta*, (105):40–52, 2013.
- [14] Joan Carles Ambrojo. Los drones se alistan al servicio civil. *Técnica industrial*, (303):18–19, 2013.
- [15] Belén Melián, José A Moreno Pérez, and J Marcos Moreno Vega. Metaheurísticas: Una visión global. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 7(19):0, 2003.
- [16] James P Keliy. Meta-heuristics: Theory & applications. 1996.
- [17] MA Gutiérrez Andrade, S De los Cobos Silva, and Blanca Rosa Pérez Salvador. Optimización con recocido simulado para el problema de conjunto independiente. *Revista En Línea, México, vol. Época II*, (3):27–50, 1995.
- [18] María D Torres Soto, Aurora Torres Soto, Jaime R de la Torre Sifuentes, Eunice E Ponce de León Sentí, and Francisco J Luna Rosas. Estudio comparativo de algoritmos de búsqueda local.
- [19] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549, 1986.

-
- [20] Adenso Díaz, Fred Glover, Hassan M Ghaziri, JL González, Manuel Laguna, Pablo Moscato, and Fan T Tseng. Optimización heurística y redes neuronales, 2000.
 - [21] Alicia Riojas Cañari and María Álvarez Rivas. Aplicación de la metaheurística búsqueda tabú al problema de las n-reinas. *Revista de investigación de Sistemas e Informática*, 6(2):27–34, 2009.
 - [22] Fred Glover and Belén Melián-Batista. Búsqueda tabú. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 7(19):29–48, 2003.
 - [23] Mario César Vélez and José Alejandro Montoya. Metaheurísticos: una alternativa para la solución de problemas combinatorios en administración de operaciones. *Revista Eia*, (8):99–115, 2007.
 - [24] Wilson Parra Ardila and Ricardo Andrés Cárdenas Izquierdo. Estudio de la métrica de manhattan: segmentos, rectas, rayos, circunferencias y algunos lugares geométricos en la geometría del taxista. 2013.
 - [25] Vera Mersheeva and Gerhard Friedrich. Multi-uav monitoring with priorities and limited energy resources. In *ICAPS*, pages 347–356, 2015.

FICHA AUTOBIOGRÁFICA

Deiby Andrés Parra Gaviria

Candidato para el grado de Maestría en Ciencias en
Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

SISTEMA MULTIAGENTE PARA EL MANEJO
ÓPTIMO DE UN CULTIVO EMPLEANDO LA
METODOLOGÍA PROMETHEUS

Nací en el municipio de Ansermanuevo (Colombia), un 24 de julio, misma fecha del nacimiento del libertador Simón Bolívar, hecho que siempre ha llevado a mi madre a decirme que nací para grandes victorias y pocos amigos. Desde pequeño siempre quise viajar por el mundo, lo que me trajo a México a continuar mi formación académica.

En los sitios donde he estado, he hecho grandes amigos (con lo cual no se cumplió la profecía de mi madre) y he podido conocer la calidez de la amistad y la camaradería. De estos grandes amigos, una parte quedaron en los años mozos del colegio (Santa Ana de los Caballeros) y la otra en la época universitaria, en una

ciudad hermosa que me acogió como un hijo más y fue testiga de mi transición de adolescente rebelde a un hombre de bien: mi querida Pereira.

En esta gran tierra, culminé mis estudios en Ingeniería Industrial (Universidad Tecnológica de Pereira), aprendí a cocinar y a ser independiente (lo que ha sido de gran utilidad) y me embarque en la aventura de continuar mi crecimiento personal y profesional, siempre con la venia de mi madre y mil sueños por cumplir.

Llegué a Monterrey con mi mochila y mis objetivos, encontré más amigos y personas que me hacen sentir como en mi casa y luego de dos años de clases, del afán de las mañanas y exámenes, culmino este objetivo con el trabajo: *Sistema multiagente para el manejo óptimo de un cultivo empleando la metodología Prometheus*, el cual estimado lector, pongo a su disposición.